AFRL-IF-RS-TR-2002-32

Final Technical Report

March 2002

BATTLESPACE CHALLENGE PROBLEMS EVALUATION OF HIGH PERFORMANCE KNOWLEDGE BASES (HPKB) TOOLS FOR BATTLEFIELD AWARENESS AND PLANNING

Alphatech, Incorporated

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. F101/02

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2002-32 has been reviewed and is approved for publication.

APPROVED:

CRAIG S. ANKEN
Project Engineer

FOR THE DIRECTOR:

MICHAEL TALBERT, Maj., USAF, Technical Advisor
Information Technology Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	MARCH 2002	May 97 - Sep 01

4. TITLE AND SUBTITLE
BATTLESPACE CHALLENGE PROBLEMS EVALUATION OF HIGH PERFORMANCE KNOWLEDGE BASES (HPKB)FOR BATTLEFIELD AWARENESS AND PLANNING

5. FUNDING NUMBERS
C - F30602-97-C-0190
PE - 62301E
PR - IIST
TA - 00
WU - 01

6. AUTHOR(S)
Eric K. Jones, Robert R. Tenney, Kendra E. Moore,
Joel S. Douglas, and Leonard Lublin

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Alphatech, Incorporated
50 Mall Road
Burlington Massachusetts 01803

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203-1714

Air Force Research Laboratory/IFTD
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2002-32

11. SUPPLEMENTARY NOTES
Air Force Research Laboratory Project Engineer: Craig S. Anken/IFTD/(315) 330-2074

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***
This final report documents a series of mini Challenge Problem (CP) evaluations aimed at addressing key technical issues in the application of a broad range of Artificial Intelligence (AI) and related information technologies to timely battlespace problems.
The Challenge Problem process includes:
Define the problem or need addressed.
Match/validate operational requirements to the broad capabilities of AI and related information technologies, and develop vision/concept/architectural views of the problem.
Assemble materials supporting the specification of the problem and associated knowledge base, including: reference documents, subject matter experts (SME), ontology development, generation of case studies and representative scenarios.
Support the development of architectural components.
Conduct evaluations of both component technologies and end-to-end systems.
Extend prototype capabilities and manage operational demonstrations/transitions.
The Final Report is structured as a compendium of summaries of multiple challenge problems investigated during the course of this program. Each CP summary addresses the specific problem/need, objective, approach and results. The primary Challenge problems addressed under this effort included: Movement Analysis, Workarounds Reasoning, Course-of-Action Analysis, Endstate Analysis, Integrated COA Critiquing & Elaboration, and BioSurveillance.

14. SUBJECT TERMS
Knowledge Base Technology, Artificial Intelligence, Information Technology, Challenge Problem, Evaluations

15. NUMBER OF PAGES
80

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures

1. Introduction

This document provides the Final Report for the tasks conducted during the four years for the "Evaluation of HPKB Tools for Battlefield Awareness and Planning" program under Air Force Contract # F30602-97-C-0190. In keeping with the overarching themes and objectives of the contract, ALPHATECH carried out a series of mini Challenge Problem (CP) evaluations aimed at addressing key technical issues in the application of a broad range of Artificial Intelligence (AI) and related information technologies to timely battlespace problems, while maintaining relevance to DARPA programs.

The Challenge Problem process includes:

- Define the problem or need addressed (including identifying target related programs at DARPA and operational advocates)
- Match/validate operational requirements to the broad capabilities of AI and related information technologies, and develop vision/concept/architectural views of the problem
- Assemble materials supporting the specification of the problem and associated knowledge base, including: reference documents, subject matter experts (SME), ontology development, generation of case studies and representative scenarios
- Support the development of architectural components
- Conduct evaluations of both component technologies and end-to-end systems
- Extend prototype capabilities and manage operational demonstrations/transitions.

This Final Report is structured as a compendium of summaries of the eleven challenge problems investigated during the course of this four year program. Each CP summary addresses the specific problem/need, objective, approach and results. For additional details and background information, the reader is directed to the more than 2 dozen formal contract reports and additional technical memoranda/briefings previously delivered. The eleven Challenge Problems subsequently addressed in this report and their period of study are shown below. The last section of this report provides some conclusions and retrospective remarks.

Challenge Problem	Time Frame
Movement Analysis	Year 1
Workarounds Reasoning	Year 1
Course-of-Action Analysis	Year 2
LoC/S Workarounds Reasoner	Year 3/4
Endstate Analysis	Year 3/4
Integrated COA Critiquing & Elaboration	Year 3
BioSurveillance	Year 3
SHIELD	Year 3/4
TBM Reasoner	Year 3/4
JBI Study	Year 4
JBI Integration	Year 4

2. Movement Analysis

2.1 Problem

The Movement Analysis challenge problem concerns high-level analysis of idealized sensor data, particularly the airborne JSTARS' Moving Target Indicator radar. This Doppler radar can generate vast quantities of information—one reading per minute for each vehicle in motion within a 10,000 square mile area. The Movement Analysis scenario involves an enemy mobilizing a full division of ground forces—roughly 200 military units and 2000 vehicles—to defend against a possible attack. A simulation of the vehicle movements of this division was developed, the output of which includes reports of the positions of all of the vehicles in the division at one minute intervals over a four-day period, for eighteen hours each day. These military vehicle movements were then interspersed with plausible civilian traffic, to add the problem of distinguishing military from non-military traffic. The Movement Analysis task is to monitor the movements of the enemy to detect and identify types of military sites and convoys.

Because HPKB is not concerned with signal processing, the inputs are not real JSTARS data but are instead generated by a simulator and preprocessed into vehicle "tracks." There is no uncertainty in vehicle location and no radar shadowing, and each vehicle is always accurately identified by a unique "bumper number." However, vehicle tracks do not precisely identify vehicle type, but instead report each vehicle as being either light-wheeled, heavy-wheeled, or tracked. Low-speed and stationary vehicles are not reported.

Vehicle track data are supplemented by small quantities of high-value intelligence data, including accurate identification of a few key enemy sites, "electronic intelligence" reports of locations and times at which an enemy radar is turned on, "communications intelligence" reports that summarize information obtained by monitoring enemy communications, and "human intelligence" reports that provide detailed information about the numbers and types of vehicles passing a given location. Other inputs include a detailed road network in electronic form, and an order of battle that describes the structure and composition of the enemy forces in the scenario region.

2.2 Objectives

Given these inputs, Movement Analysis CP comprises the following tasks:

- Distinguish military from non-military traffic. Almost all military traffic travels in convoys, which makes this a fairly straightforward task except for very small convoys of two or three vehicles.
- Identify the sites between which military convoys travel, determine which of these sites are militarily significant, and determine the types of each militarily significant site. Site types include battle positions, command posts, support areas, air defense sites, artillery sites, and assembly/staging areas.

- Identify which units (or parts of units) in the enemy order of battle are participating in each military convoy.
- Determine the purpose of each convoy movement. Purposes include reconnaissance, movement of an entire unit towards a battle position, activities by command elements, and support activities.
- Infer the exact types of the vehicles that make up each convoy. About twenty types of military vehicles are distinguished in the enemy order of battle, all of which show up in the scenario data.

To help the technology base and the integration teams develop their systems, a portion of the simulation data was released in advance of the evaluation phase, accompanied by an answer key that supplied "model answers" for each of the inference tasks listed above.

2.3 Approach

Battlespace CP developers had a development phase of about 11 months followed by a two-week test period. Sample problems and data were provided during the development period to allow developers to build systems capable of the specified functionality. Evaluation proceeded in two phases:

- A *test phase*, whose aim was to establish a performance baseline. Problems and tasks were chosen that were similar but not identical to the sample problems.
- A *modification phase*, whose aim was to test the ability of developers to rapidly extend and modify their knowledge bases (KBs) and associate knowledge-based systems. Problems and tasks were chosen that went beyond the sample problems in significant respects.

Each phase took approximately one week.

Within each phase, movement analysis systems were given a test and a retest on the same scenario. In the test phase, simulation data for a brigade-level scenario originally provided as part of the sample data were augmented by data for an additional four brigades. In the modification phase, these data were further extended to include reports generated from certain rarely occurring but high value patterns, including behaviors of SCUD and LBCM units.

The data for the largest simulation (the simulation for the modification phase) covered an area of approximately 10,000 square kilometers and a period of approximately four simulated days. The scenario involved 194 military units, 1848 military vehicles, and 7666 civilian vehicles. Some units were co-located at sites, others stood alone. There were 726 convoys and nearly 1.8 million distinct reports of vehicle movements.

At each phase of the evaluation, participants were provided with two data sets: a data set containing only military traffic, and a data set containing reports of civilian traffic in addition to the military traffic. They were encouraged to submit results for both data

sets, to help determine the extent to which civilian traffic "noise" would impair performance.

Four groups developed systems for all or part of the movement analysis problem, and SAIC and Teknowledge provided integration support. The groups were Stanford's Section on Medical Informatics (SMI), SRI, the University of Massachusetts (UMass), and MIT.

The following aspects of the challenge problem were scored during the evaluation:

- Site identification: identify militarily significant sites
- Site classification: determine the type (function) of an identified site
- Convoy identification: identify convoys of military vehicles

Many other aspects could have usefully been scored, but program resources and schedules did not permit further scoring and analysis.

Each site identification was scored for its accuracy, and recall and precision scores were maintained for each site. Suppose an identification asserts at time t that a battalion command post exists at a location (x,y). To score this identification, we find all sites within a fixed radius of (x,y). Some site types are very similar to others; for example, all command posts have similar characteristics. So if one mistakes a battalion command post for, say, a division command post, then one should get partial credit for the identification. The identification is incorrect, but not as incorrect as, say, mistaking a command post for an artillery site. *Entity error* ranges from zero for completely correct and maximally detailed identifications to one for hopelessly wrong identifications. Fractional entity errors provide partial credit for "near miss" identifications—identifications that were incorrect in some but not all respects, or that were correct but insufficiently detailed. If one of the sites within the radius of an identification matches the identification (e.g., a battalion command post) then the identification score is zero, but if none matches, then the score is the average entity error for the identification matched against all the sites in the radius. If no site exists within the radius of an identification, then the identification is a *false positive*.

Recall and precision rates for sites are defined in terms of entity error. Let H be the number of sites identified with zero entity error, M be the number of sites identified with entity error less than one (near misses), and R be the number of sites identified with maximum entity error. Let T be the total number of sites, N be the total number of identifications, and F be the number of false positives. The following statistics describe the performance of the movement analysis systems:

zero entity error recall = H/T

non-one entity error recall = (H+M)/T

maximum entity error recall = (H+M+R)/T

zero entity error precision = H/N

non-one entity error precision = (H+M)/N

maximum entity error precision = (H+M+R)/N

false positive rate = F/N

The experiment design involved two phases with a test and retest within each phase. Additionally, the datasets for each test included either military traffic only or military plus civilian traffic. Had each group run their systems in each experimental condition there would have been 4 tests, 2 datasets with 2 versions of each (military traffic only and military plus civilian), and 4 groups, or 64 conditions to score. All these conditions were not run. Delays were introduced by the process of reformatting data to make it compatible with a scoring program, so scores were not released in time for groups to modify their systems; one group devoted much time to scoring and did not participate in all trials; another group participated in no trials for reasons discussed at the end of this section.

2.4 Results

The trials that were run are summarized in Figure 2.1. Trials labeled "A" are from the first phase of the experiment, before the scenario modification. Trials labeled "B" are from after the scenario modification. "Mil" and "Mil+Civ" refer to datasets with military traffic only and with both military and civilian traffic, respectively. Recall rates range from 40% to just over 60%, but these are for maximum entity error recall—the frequency of detecting a site when one exists, but not identifying it correctly. Rates for correct and near miss identification are lower, ranging from 10% to 15%, and are not shown in Figure 2.1. Of the participating research groups, MIT did not attempt to identify the types of sites, only whether sites are present, so their entity error is always maximum. The other groups did not attempt to identify all kinds of sites; for example, UMass tried to identify only battle positions, command posts and assembly/staging areas. Even so, each group was scored against all site types. Obviously the scores would be somewhat higher had the groups been scored against only the kinds of sites they intended to identify (e.g., recall rates for UMass ranged from 20% to 39% for the sites they tried to identify).

Precision rates are reported only for the SMI and UMass teams, as MIT did not try to identify the types of sites. UMass's precision was highest on their first trial; a small modification to their system boosted recall but at a significant cost to precision. SMI's precision hovered around 20% on all trials.

Scores were much higher for convoy detection. While scoring convoy identifications posed some interesting technical challenges, the results were plain enough: SMI, UMass and MIT detected 507 (70%), 616 (85%) and 465 (64%) of the convoys, respectively. The differences between these figures do not indicate that one technology was superior to another because each group emphasized a slightly different aspect of the problem. For example, SMI didn't try to detect small convoys (fewer than four vehicles). In any case, the scores for convoy identifications are quite similar.

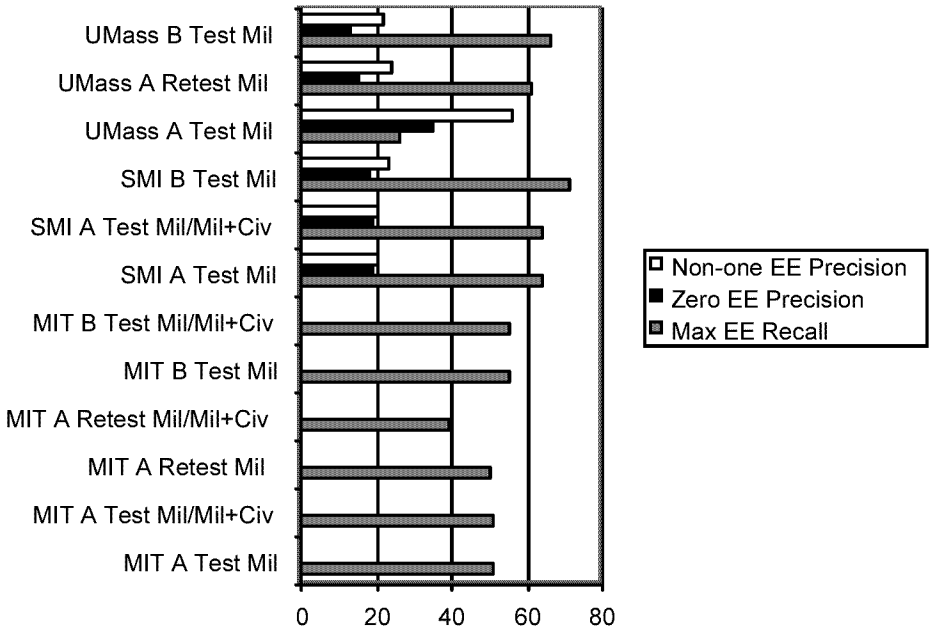

Figure 2.1. Movement Analysis Trials.

None of the systems classified site types accurately, although all detected sites and convoys fairly well. The reason for these results is that sites can be detected by observing vehicle halts, just as convoys can be detected by observing clusters of vehicles. But it is difficult to identify site types without more information. It would be worth studying the types of information that humans require for the task. No groups attempted to extend their systems to classify the new types of sites that were present in the modification data.

3. Workarounds Reasoning

3.1 Problem Overview

The Workarounds CP supports air campaign planning by the Joint Forces Air Component Commander (JFACC) and his staff. One task for the JFACC is to determine suitable targets for air strikes. Good targets allow one to achieve maximum military effect with minimum risk to friendly forces and minimum loss of life on all sides. Infrastructure often provides such targets: It can be sufficient to destroy supplies at a few key sites or critical nodes in a transportation network, such as bridges along supply routes. However, bridges and other targets can be repaired, and there is little point in destroying a bridge if an available fording site is nearby. If a plan requires an interruption in traffic of several days, and the bridge can be repaired in a few hours, then another target might be more suitable. Target selection, then, requires some reasoning about how an enemy may "work around" the damage to the target.

3.2 Objective

The task of the Workarounds Reasoning challenge problem is to automatically assess how rapidly and by what method an enemy can reconstitute or bypass damage to a target, and thereby to help air campaign planners rapidly choose effective targets. The Workarounds task involves detailed representation of targets and the local terrain around the target, and detailed reasoning about actions the enemy can take to reconstitute or bypass this damage. The inputs to Workarounds Reasoning systems include:

- A description of a target (e.g. a bridge or a tunnel), the damage to it (e.g. one span of a bridge is dropped; the bridge and vicinity are mined), and key features of the local terrain (e.g. the slope and soil types of a terrain cross-section coincident with the road near the bridge, together with the maximum depth and speed of any river or stream the bridge crosses).
- A specific enemy unit or capability to be interdicted, such as a particular armored battalion, or supply trucks carrying ammunition.
- A time period over which that unit or capability is to be denied access to the targeted route. The presumption is that the enemy will try to repair the damage within this time period; a target is considered to be effective if there appears to be no way for the enemy to do this.
- A detailed description of the enemy resources in the area that could be used to repair the damage. For the most part, repairs to battle damage are carried out by Army engineers, so this description takes the form of a detailed engineering order of battle.

All inputs are provided in a formal representation language.

3.3 Approach

The workarounds generator is expected to provide three outputs. First, a *reconstitution schedule* giving the capacity of the damaged link as a function of time since the damage was inflicted. For example, the workarounds generator might conclude that the capacity of the link is zero for the first 48 hours, but thereafter a temporary bridge will be in place that can sustain a capacity of 170 vehicles per hour. Second, a *time line of engineering actions* that the enemy might carry out to implement the repair, the time these actions require, and temporal constraints among them. If there appears to be more than one viable repair strategy, a time line should be provided for each. Third, a *set of required assets*: For each time line of actions, a description of the engineering resources that are used to repair the damage, and pointers to the actions in the time line that employ these assets. The reconstitution schedule provides the minimal information required to evaluate the suitability of a given target. The time line of actions provides an explanation to justify the reconstitution schedule. The set of required assets is easily derived from the time line of actions, and can be used to suggest further targets for pre-emptive air strikes against the enemy to frustrate its repair efforts.

A training data set was provided to help CP developers build their systems. It supplied inputs and outputs for several sample problems, together with detailed descriptions of the calculations carried out to compute action durations, lists of simplifying assumptions made to facilitate these calculations, and pointers to text sources for information on engineering resources and their use (mainly Army Field manuals available on the World-Wide Web).

Battlespace CP developers had a development phase of about 11 months followed by a two-week test period. Sample problems and data were provided during the development period to allow developers to build systems capable of the specified functionality. Evaluation proceeded in two phases:

- A *test phase*, whose aim was to establish a performance baseline. Problems and tasks were chosen that were similar but not identical to the sample problems.
- A *modification phase*, whose aim was to test the ability of developers to rapidly extend and modify their knowledge bases (KBs) and associate knowledge-based systems. Problems and tasks were chosen that went beyond the sample problems in significant respects.

Each phase took approximately one week.

Within each phase, workarounds systems were given a test and a retest on the same problems. In the first phase, the systems were tested on twenty problems and re-tested after a week on the same problems. In the second phase, a modification was introduced into the scenario, the systems were tested on five problems, and after a week re-tested on the same five problems and five new ones.

Solutions were scored along five equally-weighted dimensions:

1. Viability of enumerated workaround options
2. Correctness of the overall time estimate for a workaround
3. Correctness of solution steps provided for each viable option
4. Correctness of temporal constraints among these steps
5. Appropriateness of engineering resources employed.

Scores were assigned by comparing the systems' answers with those of human experts. Bonus points were awarded when, occasionally, systems gave better answers than the experts. These answers became the gold standard for scoring answers when the systems were re-tested.

3.4 Results

Four systems were fielded by ISI, George Mason University, Teknowledge, and AIAI (Edinburgh). The results are summarized in Figures 3.1 and 3.2 for the Initial and Modification Phases, respectively. Lighter bars represent test scores, darker bars, retest scores. Circles represent the "scope" of the task attempted by each system, i.e., the best score that the system could have received given the number of questions it answered. (Note that for the Modification Phase, only five problems were released for the test, ten for the re-test, so the maximum available points for each are 50 and 100, respectively.) In each figure, the upper extreme of the vertical axis represents the maximum score a system could get by answering all the questions correctly (i.e., 200 points for the initial phase, 50 points for the first test in the modification phase, 100 points for the re-test in the modification phase). The bars represent the number of points scored, and the circles represent the number of points that could have been awarded given the number of questions that were actually answered. For example, in the initial phase, ISI answered all questions so could have been awarded 200 points on the test and 200 on the re-test; GMU covered only a portion of the domain and it could have been awarded a maximum of 90 and 100 points, respectively. The bars represent the number of points actually scored by each system.

How one views the performance of these systems depends on how one values correctness, coverage (the number of questions answered) and, more subtly, the prospects for scaling the systems to larger problem sets. An assistant should answer any question posed to it, but if the system is less than ideal should it answer more questions with some errors or fewer questions with fewer errors? Obviously the answer depends on the severity of errors and on the application, on the prospect for improving system coverage and for improving accuracy. What we might call "errors of specificity," in which an answer is less specific or complete than it should be, are not inconsistent with the philosophy of HPKB, which expects systems to give partial—even common sense—answers when they lack specific knowledge.

Figures 3.1 and 3.2 show that ISI was the only group to attempt to solve all the workaround problems, although its answers were not all correct; whereas GMU solved fewer problems with higher overall correctness. AIAI solved fewer problems still, but

quite correctly, and Teknowledge solved more problems than AIAI with more variable correctness. One can compute coverage and correctness scores as follows: Coverage is the number of questions attempted divided by the total number of questions in the experiment (55 in this case). Correctness is the total number of points awarded divided by the number of points that might have been awarded given the number of answers attempted. Figure 3.3 shows a plot of coverage against correctness for all the workaround systems. Points above and to the right of other points are superior, thus, the ISI and GMU systems are preferred to the other systems, but the ranking of these systems depends on how one values coverage and correctness.

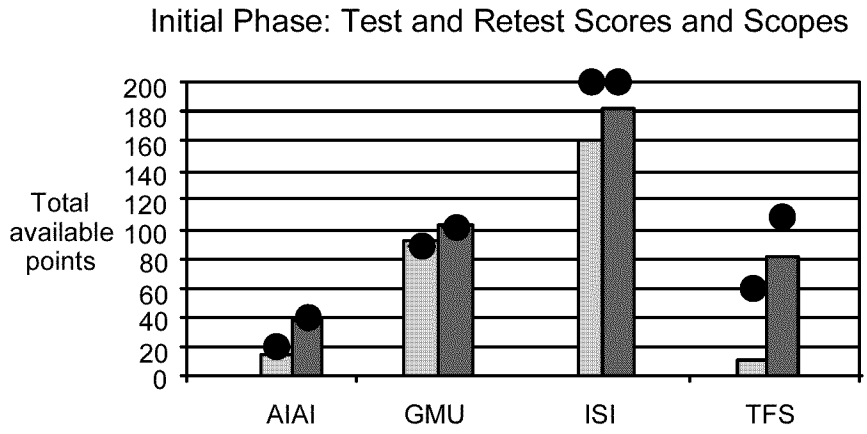

Figure 3.1. Results of the initial phase of the workaround evaluation.

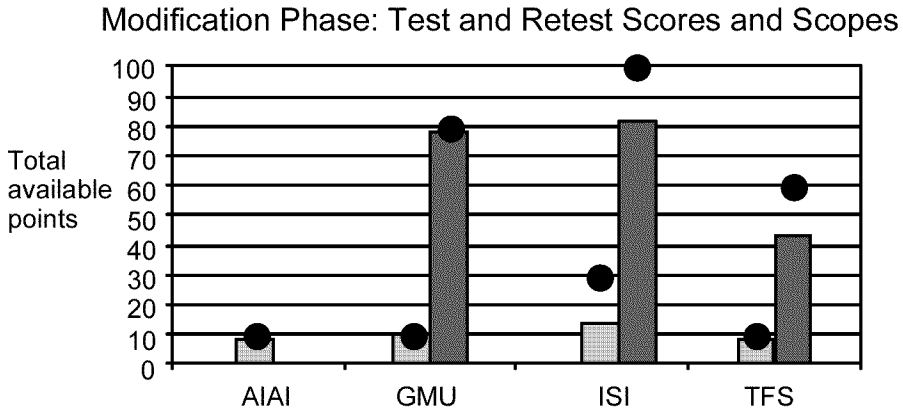

Figure 3.2. Results of the modification phase of the workaround evaluation.

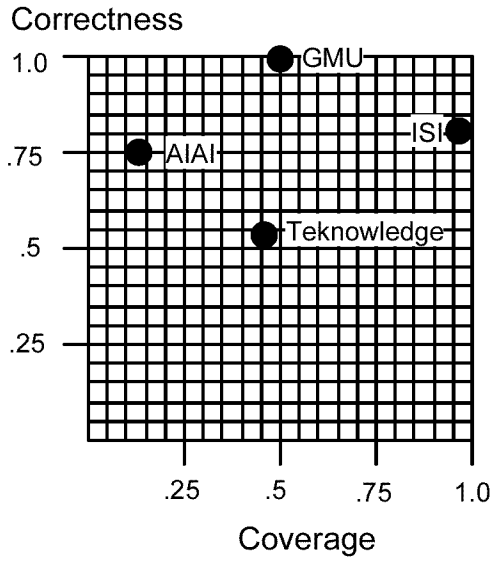

Figure 3.3. Plot of overall coverage against overall correctness for each system.

4. COA Analysis

4.1 Problem

The Course of Action Challenge Problem aimed to develop a knowledge-based decision support capability for Army combat planners. An Army planning staff and its commander construct plans using a well-specified process called the *military decision-making process* (MDMP). *Courses of action* (or COAs) are central to the MDMP. A COA is a sketchy plan of action for a military unit. Through the various steps of the MDMP, planners propose one or more friendly COAs, and then assess these COAs in depth, at the same time fleshing out the plans they embody to provide considerably greater detail than the original COA.

The Army describes courses of action using *COA statements* and *COA sketches*. A fully specified course of action consists of *both* a COA statement and a sketch. While both forms convey elements of the same concept, neither a sketch nor a statement can fully communicate the concept alone. It is interesting to note, however, that both statement and sketch, taken separately, should be sufficient to convey much of the meaning in the other element. A competent military planner, for example, should be able to look at only the sketch and state, using precise terminology, much of the contents of the paragraph. It is the formal nature of the military's practice, vernacular, and symbology that makes this possible, giving us much leverage in understanding the domain.

Assessing a COA and fleshing out the plan it embodies are intimately related to one another. COA analysis involves war-gaming friendly COAs against a number of possible enemy COAs, thereby inferring many details of the plan that only become explicit through a detailed simulation. Planners keep a record of these details; this record is then incorporated into the *operations order*, the final and most fully detailed version of the plan. The detailed information that war-gaming produces often exposes flaws in a COA; COA analysis also involves repairing those flaws that can be fixed. If a sufficiently serious flaw is found that cannot be fixed, the COA must be abandoned.

4.2 Objective

The objective of the COA CP task was to build a system that could help planners (a) determine whether or not a COA is viable, (b) assess the strengths and weaknesses of COAs, (c) suggest ways to improve a flawed COA, and (d) further develop the plan a COA embodies. The system was intended to support both the COA development and COA analysis steps of the MDMP, with a focus mainly on the latter. The COA CP system *was not* intended to be a complete operational system for assessing and further developing COAs, but *was* supposed to address many of the hard problems of knowledge-based reasoning that such an operational system would face.

4.3 Approach

A "parameterized questions" approach (similar to that adopted in the Year 1 crisis management CP) was employed to specify the inputs to the CP task. Such parameterized

questions allowed the scope of inputs to the CP to be precisely specified, and were useful in evaluating the system's performance.

The inputs to the COA CP involved more than just parameterized questions, however. Military planners specify COAs and the required scenario elements using a mix of sketching and natural language. The COA CP employed input formats conforming as closely as possible to those employed by military planners. Inputs were specified using a combination of sketches and natural language, using a set of sketch elements and a restricted English grammar defined for the COA CP.

Answers to a variety of sample questions were used to specify the outputs of the CP task. In addition to the answers themselves, detailed justifications were required to be supplied for each answer.

The following elements made up the inputs to the COA CP task:

1. Scenarios, encoded in terms of the products of mission analysis (PMA),
2. COA sketches, COA statements, or both, addressing the mission of the scenarios,
3. Assumptions made for planning purposes, and
4. Questions about the COA that the sketch and/or statement defines.

The output of the CP task was defined as answers to these parameterized questions and justifications for the answers. The COA CP task was decomposed into a number of subtasks, identified as both relevant to needs of military planners and suited to the goals and capabilities offered by HPKB. Each parameterized question in the input to the CP addressed one of the subtasks identified below:

1. Assess the viability of a course of action (and where possible suggest repairs for flaws that are discovered) in terms of Suitability, Feasibility, Acceptability, and Completeness.
2. Assess the correctness of a course of action (and where possible suggest repairs for flaws that are discovered), including the correctness of its Array of Forces, Scheme of Maneuver, and Command and Control.
3. Identify the strengths and weaknesses of a course of action with respect to established abstract principles (and where appropriate suggest repairs for weaknesses that are identified), including assessment of the extent to which the course of action exemplifies the Principles of War and/or the Tenets of Army Operations.
4. Further analyze and develop the course of action, including specifying any required branch plans, analyzing or refining the location and timing of the decisive point, estimating the duration of each critical event in the COA, and analyzing or refining the location and timing of each decision point.
5. Perform enabling tasks, such as identifying elements in the COA statement, COA sketch, or products of mission analysis; inferring spatial relations between elements in a COA sketch and/or a battlefield effects sketch, estimating mobility of a force as

function of terrain, and estimating force ratios, attrition, and the time required to perform a task.

Developer teams with critiquing tools to be evaluated included: the Learning Agents Laboratory at George Mason University (GMU); the Expect Group at the University of Southern California (USC) Information Sciences Institute (ISI); the Loom/PowerLoom Group at USC/ISI; and a joint team of Teknowledge, Inc. and Cycorp, Inc. The Experimental Knowledge Systems Laboratory at the University of Massachusetts (UMass) developed a tool to dynamically simulate the interaction of friendly and enemy courses of action.

Teknowledge, Inc. was the integration contractor for the COA CP Challenge Problem, supported by Cycorp, Inc., Science Applications International Corporation (SAIC), the University of Edinburgh's Artificial Intelligence Applications Institute (AIAI), Northwestern University's Institute for the Learning Sciences (ILS), USC/ISI, and Stanford Medical Informatics (SMI).

The COA Challenge Problem results were evaluated using a process based upon test questions relating to several scenarios, as follows:

1. *Baseline:* employ PMA and one or more friendly COAs drawn from the sample scenarios provided as part of the sample scenario materials, together with sample questions also drawn from these sample scenario materials.

2. *Baseline with new test questions:* employ same PMA and friendly COAs as in 1, but include test questions that were not addressed in the sample scenario materials.

3. *Baseline with new friendly COA and test questions:* employ PMA from the same baseline scenarios as in 1, but provide new friendly COAs to critique. Test questions will be drawn from the entire space of parameterized questions.

4. *Minor variant scenario:* employ one of the same scenarios as in 1, 2, and 3 but with a change in mission. PMA are unchanged except as affected by change in mission, but new friendly COAs will be provided. Test questions will be drawn from the entire space of parameterized questions.

5. *New scenario:* employ all new PMA and friendly COAs. Test questions will be drawn from the entire space of parameterized questions.

Following a dry run period to debug the mechanics of the evaluation process, the final evaluation was accomplished during a two-week period, divided into two one-week cycles. Questions in groups 1 through 4 above were tested the first week, questions in group 5 the second week.

Each testing week involved the following basic process:

1. Release evaluation materials (ALPHATECH)

2. Debug shared input representations (Teknowledge, AIAI, ALPHATECH)
3. Generate initial system responses (All critiquing teams)
4. Issue model answers (ALPHATECH)
5. Repair phase (All teams)
6. Generate revised system responses (All critiquing teams).

Following the submission of all the answers for both testing weeks, the evaluation scoring process commenced.

Scoring Criteria and Metrics

To evaluate the answers submitted to the test questions, the following scoring criteria were used:

Correctness (50%) – Matches model answer or is otherwise judged to be correct.

Justification (30%) – Scored on presence, soundness, and level of detail.

Lay Intelligibility (10%) – Degree to which a lay observer can understand the answer and the justification.

Sources (10%) – Degree to which appropriate sources are noted.

Proactivity (10% extra credit) – Appropriate corrective actions or other information suggested to address a critique (i.e., an answer identifying a deficiency in the COA).

Quantitative metrics of overall performance on the critiquing task were developed by substituting different linear combinations of the scoring criteria developed in previous section for **Score** in the "metric schemas" shown in Figure 4.1. Each chosen linear combination of scoring criteria yields a different concrete metric.

- Overall % = $\dfrac{\text{Score for All Correct or Partly Correct Answers}}{\text{Number Of Original Model Answers}}$
- Recall = $\dfrac{\text{Score for Correct or Partly Correct Answers to Critiquing Questions}}{\text{Number of Original Model Answers}}$
- Precision = $\dfrac{\text{Score for All Answers to Critiquing Questions}}{\text{Number of System Answers Provided}}$

Figure 4.1. Metric schemas for HPKB COA CP

4.4 Results

Evaluating the COA Critiquers

Figure 4.2 summarizes how the evaluators classified the set of all submitted challenge problem answers.

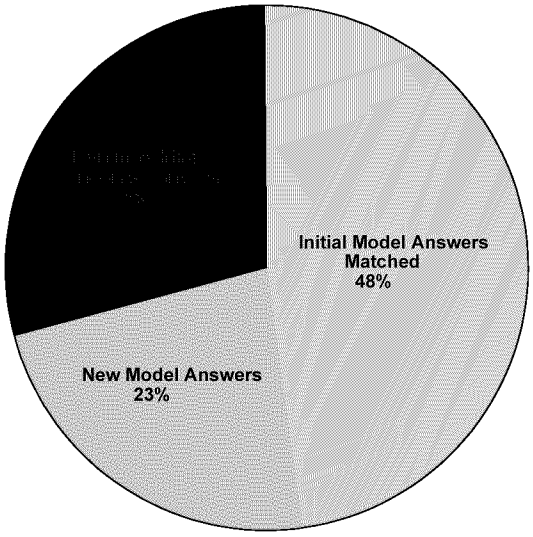

Figure 4.2. Classification of answers from all developers.

Figure 4.3 depicts the number of test questions answered by each developer during the evaluation. In the interest of providing a coherent view of the best and final results from each team, the numbers depicted in this figure include only the answer submissions to test questions for Evaluation Items 1 and 2, and the post-repair submissions for Evaluation Items 3 through 5.

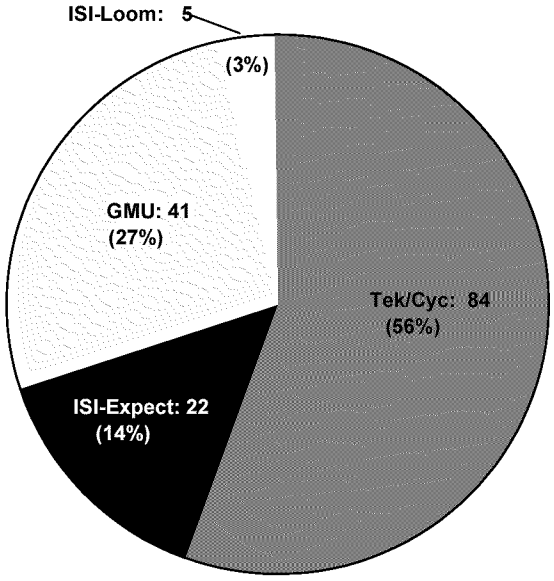

Figure 4.3. Number of test questions answered by each developer.

Figure 4.4 depicts the number of model answers generated by each of the developer teams. The numbers shown include both answers that match the original model answer and answers that were judged to be new model answers.

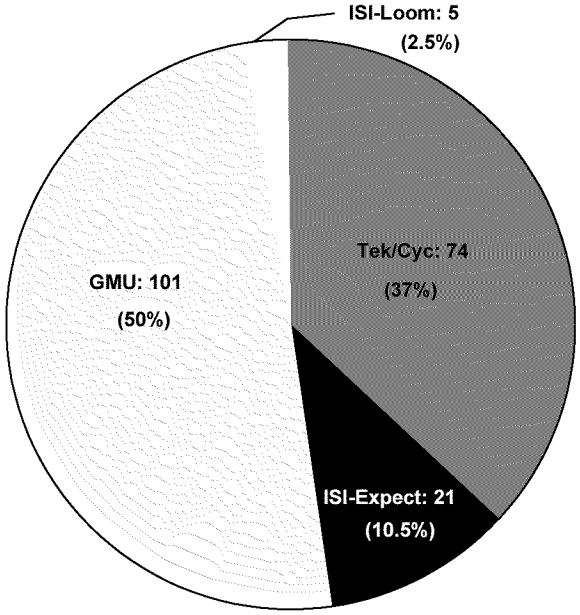

Figure 4.4. Number of model answers generated by each developer

Figure 4.5 shows the Overall % metric computed on the basis of the total score function. GMU's Disciple critiquer scored highest for this metric by a significant margin. Because of the open-ended nature of the strengths and weaknesses critique questions addressed by the GMU system, and the definition of the metric (counting scores for all correct answers including new model answers, but dividing only by the number of original model answers), it was possible to exceed 100%.

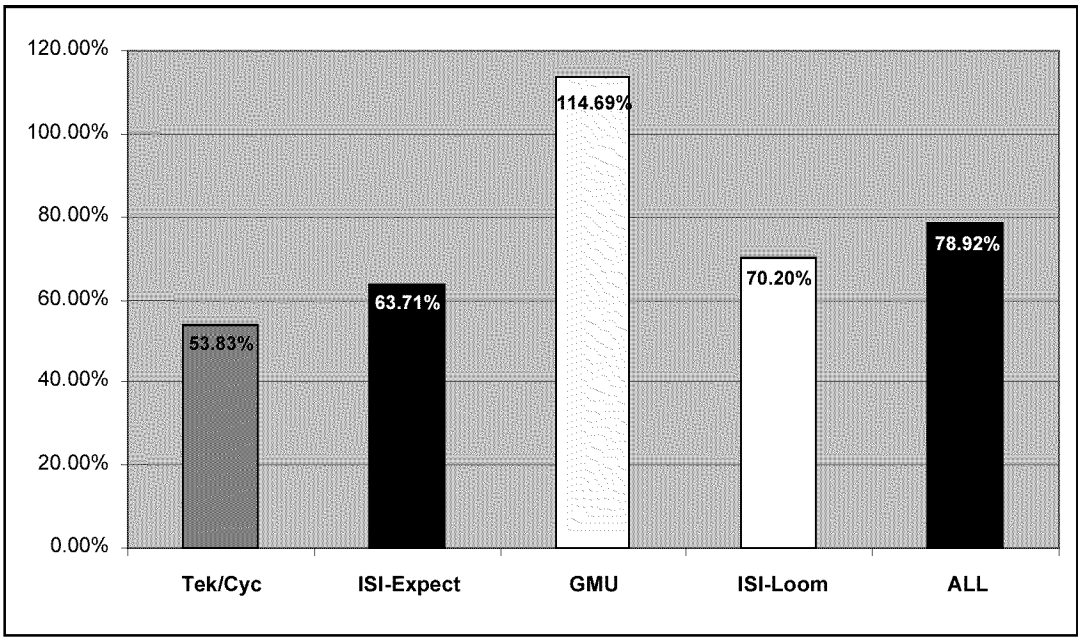

Figure 4.5. Metric: Overall % (computed on Total Score)

Evaluating the UMass Simulator

The COA simulator developed by the Experimental Knowledge Systems Laboratory at the University of Massachusetts was evaluated separately from critiquers. The evaluation was less formal than for the COA critiquing tools and involved no numeric scoring. This was because both the problem inputs and the published scoring criteria were designed around the concept of critiquing rather than simulation. The goals of this evaluation were to validate the simulator's design assumptions, to identify and analyze any surprising results, and to provide feedback to guide further development. The evaluators selected a single scenario (scenario 3), and engaged the UMass development team in an iterative debugging process during which the evaluation team enhanced and extended the scenario inputs where they were found to be lacking, while the UMass team worked to correct any incorrect simulator behavior identified by the evaluators.

The evaluators identified a number of simulator behaviors that needed improvement or correction. For instance, implementations of various tactical tasks revealed aspects where additional specification was needed, including:

1) The "passage of lines" task, in which rate of movement slowed excessively during passage of one unit through another, due to method of imposing effects of friction between adjacent units,

2) The "block" task, where units lined up defensively would all attempt to move at an approaching unit and attack it, thus leaving their original positions undefended (and unblocked),

3) Army Aviation units tasked to strike in a particular place tended to attack anything in their range on their way to the target, and then to loiter over the target area attacking any units venturing into range.

On the other hand, this same evaluation process made it clear that the scenario inputs designed for COA critiquing lack certain information needed to properly simulate the execution of the COA. For instance:

1) COAs do not fully specify unit locations or activity timing, etc.,

2) Activities of other units outside area of operations are not adequately specified, though they can be significant to the activities within the area of operations.

Even with the weaknesses described above, the UMass COA simulator demonstrated a number of valuable results. It showed that it can provide validation or enhancement of planning assumptions:

- The "Red Most Dangerous" COA was confirmed to be a considerable threat;
- The Blue counterattack must begin well before Red forces reach OBJ GRANT or the commander's expressed intent cannot be satisfied.

The results of the simulation also demonstrated that it can indicate likely areas for elaboration of the Products of Mission Analysis and the Course of Action. For instance, in the scenario evaluated, it becomes clear that the penetration needs to happen quickly, or Red force will make too much progress before Blue's Main Effort can pass through and attack it. This is suggestive of an enhancement to the commander's guidance.

5. LoC/S Workarounds Reasoner

5.1 Problem Definition

The LoC/S Workarounds Reasoning Challenge Problem focussed on automated reasoning about multiple outages, a crucially important enhancement to existing HPKB workaround systems. Requirements analysis for the Workaround Reasoner were derived from a meeting with a prospective end-user, in which the operational use of the system was discussed in detail. In this meeting, two distinct ways of using the system were proposed:

1. During deliberate planning, to develop a detailed understanding of possible workaround options in response to individual outages affecting a particular link in the road network. For purposes of this analysis, the full range of engineering equipment typical of a country of interest may be considered as resources for reconstitution and bypass construction. This equipment may be specified to the system by listing particular equipment types or by describing an engineering force structure.

2. During crisis action planning, to evaluate the potential for reconstituting an entire network in service of accomplishing some objective, by working around multiple outages. It is anticipated that crisis action planning will build on the results of previous analyses conducted during deliberate planning. This kind of objective is best thought of as a proxy for the following two more sophisticated kinds of objectives:

   - *Objectives to minimize delay.* Repair a subset of outages to enable transportation of a pre-specified amount of commodity from a single source to a single sink as quickly as possible. In the context of natural disaster mitigation, for example, objectives of this kind arise in situations where an important "one off" rescue effort must be conducted as quickly as possible.

   - *Objectives to minimize the time required to reconstitute a given flow capacity.* Here, in contrast to minimizing "one-off" delay, the objective is to restore a given level of flow between a collection of sources and a collection of sinks as quickly as practicable. In the context of natural disaster mitigation, for example, objectives of this kind can arise in situations where it is important to continuously supply a significant volume of food and equipment to sustain a given level of disaster relief aid over an extended period.

5.2 Objectives

The objective of the LoC/S Workaround Reasoner Challenge Problem was to develop an operational prototype that either meets or demonstrates significant progress towards meeting the requirements discussed in the previous section. The implementation was to

progressively extend a pilot multiple-outage workarounds reasoning capability (called the version 0.2 workarounds reasoning system). The building of the system would also address the following problems relating to HPKB programmatic objectives:

a) The development of a large knowledge base about command, control, and utilization of combat engineering equipment for implementing workarounds

b) The automatic transformation of the knowledge in this knowledge base into an operational form suitable for highly efficient automated problem solving.

c) The integration of these knowledge-based problem-solving methods with combinatorial optimization methods from Control Theory/Operations Research to achieve novel problem-solving capabilities.

5.3 Approach

Figure 5.1 summarizes information flows within the workarounds system. Functionality is color-coded as follows:

- User interfaces are depicted in bright green ovals, while their outputs are depicted using light green.
- Inputs for the single-outage reasoner (which are also outputs of a user interface) are depicted in light green.
- Components of the single outage reasoner are depicted in orange.
- Yellow is used to depict resource allocation algorithms and other new functionality required to combine the results of deliberate planning (analyses of individual outages) into multi-outage analyses for crisis action planning.
- Digitized data products (GIS data, intelligence data, and the *link network* derived from GIS data) are depicted in blue.

Implementing this architecture required the construction of a large knowledge base for single-outage workarounds computation (programmatic objective (a), above). As a precursor to it's development, a web-based Interactive Knowledge Document (IKD) was created for accumulating, visualizing, and manipulating electronic media required for generating ontologies. The knowledge base representations are well suited for knowledge acquisition and KB maintenance, but are poorly suited to efficient problem solving. Accordingly, these representations are automatically compiled into a more operational form to support efficient single-outage workarounds reasoning (meeting programmatic objective (b)).

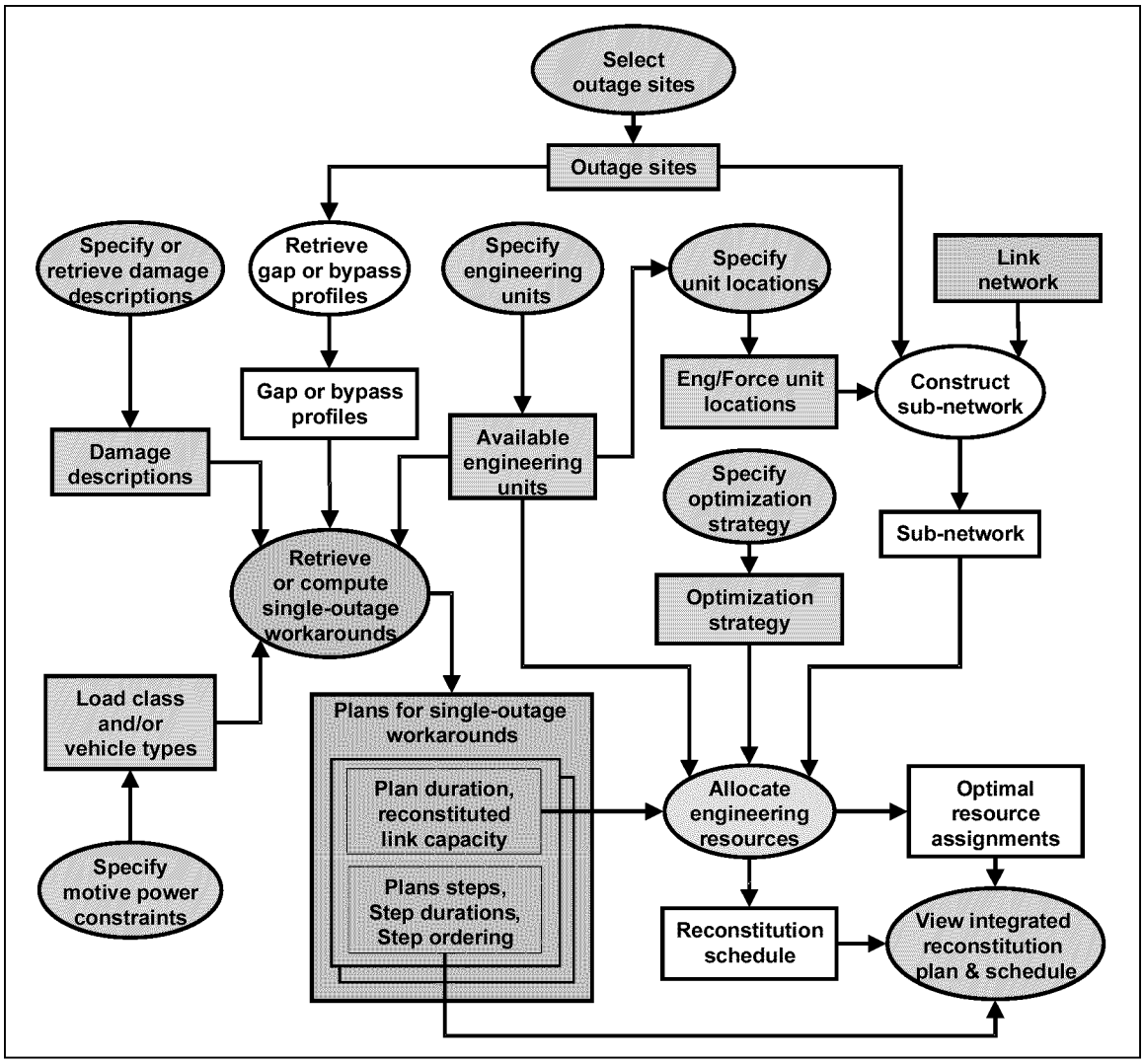

Figure 5.1. Workarounds Reasoner Information Flow Architecture

This architecture combines knowledge-based problem-solving technologies (for single-outage reasoning) and optimization technologies from Operations Research (for resource allocation) to achieve synergies unobtainable by either technology, meeting programmatic objective (c).

5.4 Results

The workarounds reasoner was developed in two spirals, extending the existing pilot capability (version 0.2) to versions 0.25 and 0.4. Version 0.25 represented a cleanup of version 0.2. It included bug fixes noted in the previous version and a more integrated user interface.

Version 0.4 utilizes the architecture depicted in Figure 5.1 to provide a decision aid for deliberate and crisis action planning. In support of deliberate planning, it allows for the investigation of the potential for bypass or reconstitution of individual outages associated with bridges or other transportation bottlenecks occurring on particular links in an

aggregated road network. It provides for exploring a range of scenarios and options by providing the following capabilities:

- Specification, lay-down, and editing of engineering equipment for reconstitution and bypass construction. The system provides a map-based display for placing engineering units in the area of interest and provides data entry forms for setting the type and quantity of equipment each unit possesses.

- Selection, semi-automatic generation, and editing of bypass gap profiles. Gap profiles are depicted graphically as transects at user-selected "gap-crossing" points, which encode key features of the banks and bed of the gap such as soil type, water width, and water depth, that may affect the ability to remedy the impediments to trafficability. Upon initial selection of a bypass gap, the underlying vector GIS database is queried for feature data for use in defining the gap. The user can edit and add data by data entry form to fully define the gap.

- Selection, specification, and editing of bridge outage characteristics, such as bridge length, width, number of spans, length of each span, and damage. Bridge outages are depicted graphically as a truss bridge profiles, showing the number of spans and damage in terms of destroyed spans. Figure 5.2 is a snapshot of the V0.4 user interface, depicting outage and gap profiles along with their associated data entry forms.

- Computation of possible workarounds. Upon execution of a single-outage analysis, the result is presented in a summary window showing a list of possible reconstitution methods organized by the kind of engineering equipment employed.

- Performing of sensitivity analysis. The user can employ the editing environment to change model inputs based upon different sets of assumptions regarding geography and bridge construction then see how they effect reconstitution.

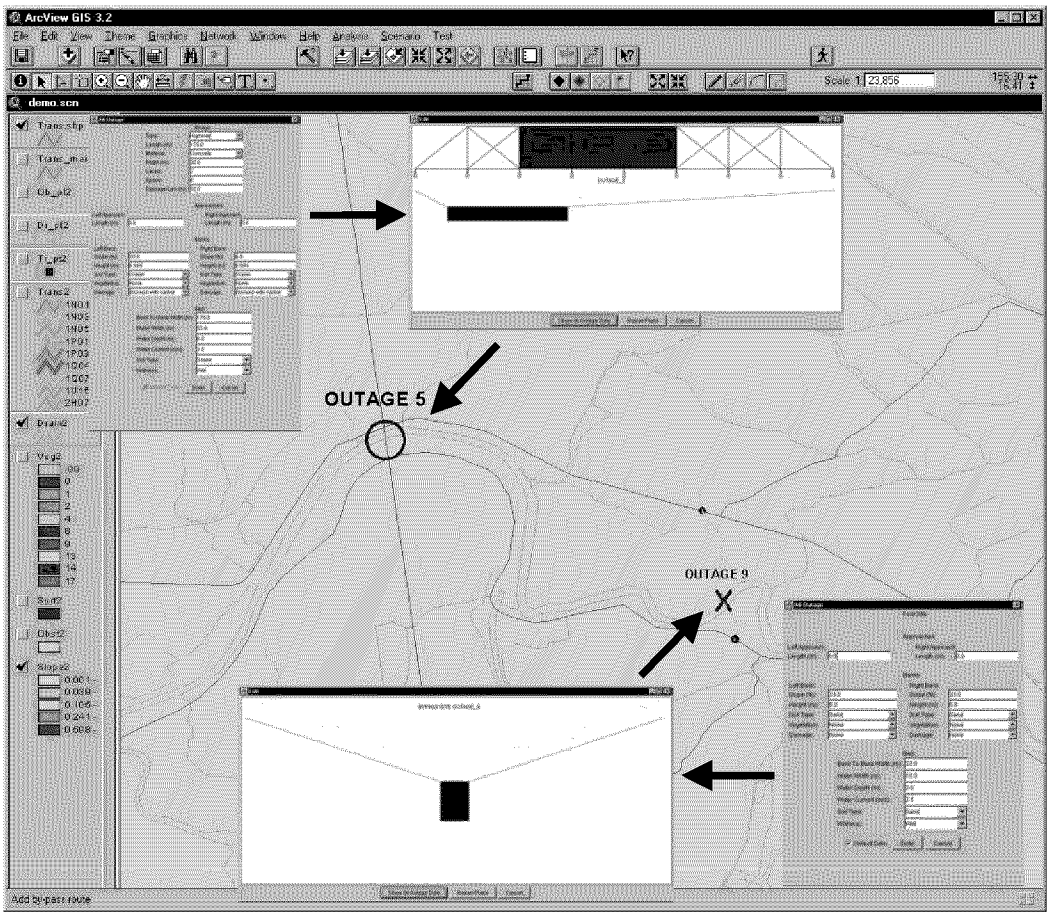

Figure 5.2. Outage and Bypass Specification

In support of crisis action planning, Version 0.4 allows for the investigation of the potential for bypass or reconstitution of multiple outages in service of some objective, relying wherever possible on the detailed single-outage analyses developed earlier during deliberate planning. In particular, the workarounds reasoner will enable analysts to rapidly explore a range of scenarios and options by providing the following capabilities:

- Specification and lay-down of a collection of sources and sinks. Sources and sinks represent the enemy's goal of the movement of commodity from location to location in minimal time. Source specification provides for entering transport vehicle speed and weight which have the effect of limiting travel time and travel route.

- Selection of a subset of outages and sinks for multi-outage analysis. The outages are those whose bridge and site characteristics presumably have been validated during deliberate planning.

- Computation of multi-outage workarounds. Upon execution of a multi-outage analysis, results are depicted graphically as a sequence of routes taken by each of the contributing engineering units and by the commodity transport vehicles along with a repair schedule presenting the optimal allocation of

engineering resources to outages for the purpose of achieving the enemy's commodity movement goal. Figure 5.3 is a snapshot of the V0.4 user interface, depicting the results of multi-outage analysis. Using the capabilities for selecting subsets of outages, sources, and sinks, the following optimization strategies are made available:

- Reconstitution of a selected subset of outages as quickly as possible.
- Reconstitution of only those outages that allow for the quickest movement of commodity (depicted in Figure 5.3).
- Both the reconstitution of a selected subset of outages and the reconstitution of those outages that allow for the quickest movement of commodity.

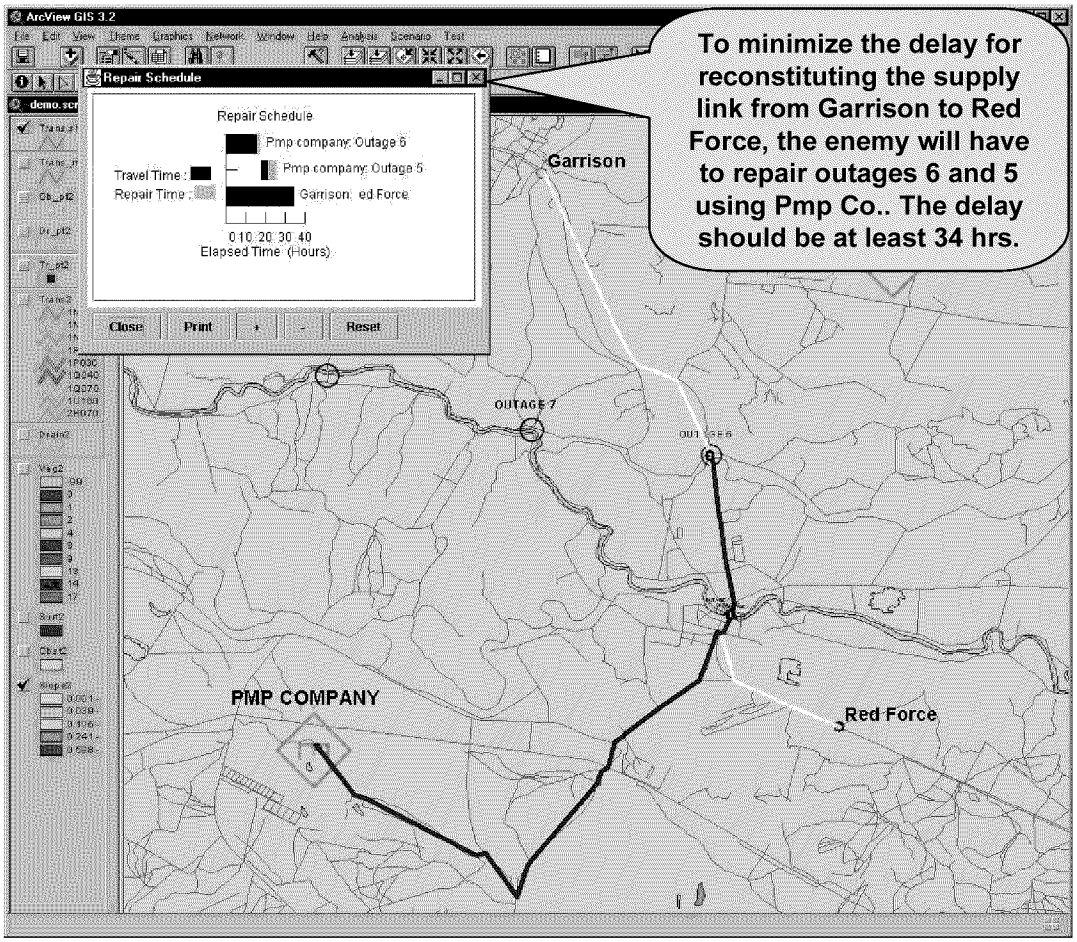

Figure 5.3. Multi-Outage Workaround Results

In order to maintain an adequate representation of the road network, Version 0.4 provides analysts with a tool suite for validating road network data, and for editing the road network to repair bugs. "Off-line" utilities are also provided for automatically transforming validated physical road network data into a form that minimizes the amount of on-line computation that must be performed to compute workarounds.

6. Endstate: Effects-Based Nonlinear Analysis and State Estimation

6.1 Statement of the Problem

Infrastructure networks provide the basis for a nation's well being in peacetime and are a pillar of strategic mobility in crisis and conflict. Conversely, understanding the weaknesses of an adversary's infrastructure and the ways in which centers of gravity depend on infrastructure networks is crucial to obtaining advantage both in crisis and in wartime.

Tools exist for modeling individual infrastructure networks. The electric power industry, the oil industry, transportation planners and others have detailed engineering models of the assets in their domain. At a higher level of abstraction, vulnerability analysis and risk assessment is becoming more common in many of these domains. However, tools that support the analysis of cross-network dependencies are still in their infancy. Development of data and models is a time-consuming and laborious process, and once constructed, models of physical networks cannot readily be correlated with real-world indications of effects. The unsolved problem of *state estimation,* or updating models from new or refined data, is still at the heart of meaningful prediction of complex system behavior. Finally, the models themselves are complicated with enormous data collection requirements and the computational complexity involved in scenario analysis with real-world data presents a formidable challenge.

6.2 Objectives

The objectives of the ENDSTATE CP were to make progress on the core technologies required to model cross-network effects in 10 component networks with 5,000 nodes in each network, with at least 20 state variables per node. As an aggregation of network models, this is a complex dynamic system with $> 10^6$ state variables. Seen from the viewpoint of the component networks, it is an aggregation of 10 dynamic systems, each with a huge number of changing boundary conditions.

Viewed from a high level, the core technology areas defined in the ENDSTATE program plan have been: (1) Cross-network effects analysis, (2) Effects-based control, and (3) State estimation. ALPHATECH's effort was concentrated on topics (1) and (2), based on the concept of model abstraction. Our research applied dynamic programming in a proof-of-concept of hierarchical control, and supported basic research by leaders in the field of control theory.

6.3 Approach

The unifying technical concept of the ENDSTATE work is *model abstraction.* ENDSTATE has considered two basic forms of model abstraction in connecting models of different systems. The first is *reduced-order modeling*, an *inductive* approach to modeling aggregated problems or sub-problems in variable spaces of reduced dimensionality. The second form of model abstraction considered is *decomposition*,

leading to hierarchies of related models, in what might be called a *deductive* approach. Useful decompositions include *geospatial* and *time-scale decomposition*. Typically, a subset of one or more independent variables in model parameter space, e.g. *(x,y,z)* location or time, is measured at varying levels of refinement in a model decomposition hierarchy.

Hypothesis

We can rapidly create and interrelate a consistent set of reduced order models (with acceptable errors) from existing system specific models.

Architectural Framework for Reduced Order Modeling

ALPHATECH's work on infrastructure networks over the years has led to the following architecture for addressing the reduced order modeling problem for multiple, interconnected networks. This architecture is rooted in a structural model abstraction hierarchy, in the FL terminology. At the lowest level, we posit the existence of *Private Agents*, or PAs. As shown in Figure 6.1 below, the PA routinely executes control decisions, denoted by {U}, upon a *Private Physical Commodity Network* (PPCN), based in part upon real-time sensor data, {SD}, generated by its own network sensors. For example, if the PA is a power utility, then its sensors generate data regarding the status and operation of its generating units, transmission lines, switches, transformer taps etc. The specific control decisions control directly the physical actuators of the network so as to deliver power according to desirable profiles. The PPCN is comprised of all physical hardware and related resources under the direct ownership and control of the PA that generates and/or transmits and/or distributes (delivers) a specific commodity or sets of commodities to its customers, located at demand nodes. The PPCN will, in general, require other "enabling commodities" for its generation and operation, which are located at the PA supply nodes.

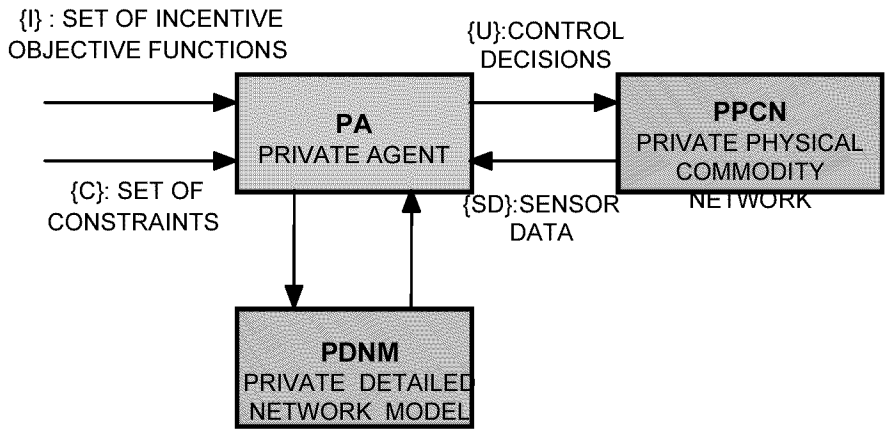

Figure 6.1. The Structure of a Private Agent (PA)

For example, if the PA is a power utility engaging in the generation and delivery of electricity, it will require oil, or gas, or coal, or hydro-resources at its generating nodes; these are examples of "enabling commodities." A limited amount of these enabling commodities may be in inventory storage, allowing for normal short-term operation in case of temporary disruptions to the supply of these enabling commodities. It will also require the availability of a communication network (often owned by the PA) to transmit data to the SCADA and transmit actuator commands. As another example, suppose that the PA is a gas delivery company. Its PPCN consists of storage tanks, pipelines, pumps etc. Clearly, electric power is an "enabling commodity" for the operation of the pipeline delivery network.

Next, we discuss the manner by which the PA determines the control decisions that will impact the operation of the PPCN. We postulate that the PA has developed *a Private Detailed Network Model* (PDNM) to be used as an internal decision aid for the development of their control decisions, {U}. In fact, many private companies have available several PDNM's, possible at different levels of fidelity and aggregation, which are used as company-private decision aids. A PDNM is a high-fidelity model of the PPCN, which is used by the PA for analysis, and evaluation of alternate control decisions. In general, such PDNM's are proprietary. Real sensor data {SD} originating from the PPCN is used to adjust the parameters and/or topology of the corresponding PDNM.

We need to capture the interdependencies and interactions of several PA's, dealing with multiple commodities, both under normal and multiple emergency conditions. In the absence of global coordinating strategies, several outages affecting multiple PA's may result in undesirable effects related to the distribution of commodities. In particular, any disruption in the normal delivery of "enabling commodities" can result in domino-like synergistic effects impacting the promised normal delivery of commodities to the consumers. For example, the destruction of a key railroad bridge may disrupt the normal delivery of coal to a power plant, which will have to be shut down after its local coal inventory is exhausted. This in turn may force the electric utility to reduce or eliminate power deliveries to a local steel plant causing disruptions in steel delivery etc. A *Coordinating Agent* (CA) is a decision-making agent whose goal is the suitable coordination of the multiple PA's under its span of control. The CA receives information from each and every PA and transmits coordination information to each and every PA.

The PA summarizes the outcome of its internal decision process by relating its current knowledge of available "enabling commodities" at its supply nodes {S} to its planned commodity deliveries to its demand nodes {D}. A PA develops a *Private Agent Aggregated Model*, a PAAM, that summarizes its delivery flows to its demand nodes {D} based upon current assumptions of required enabling commodities at its supply nodes {S}. At the simplest level, the PA simply informs the CA the assumed supply information {S} and the planned demand deliveries {D}. The detailed nature of the Private Agent Aggregated Model (PAAM) may or may not be available to the CA. For static problems the PAAM may be a multi-input multi-output nonlinear function (perhaps realized via an artificial neural network) that could be used to predict the impact of

supply changes to demand deliveries under the current operating assumptions of the PA.

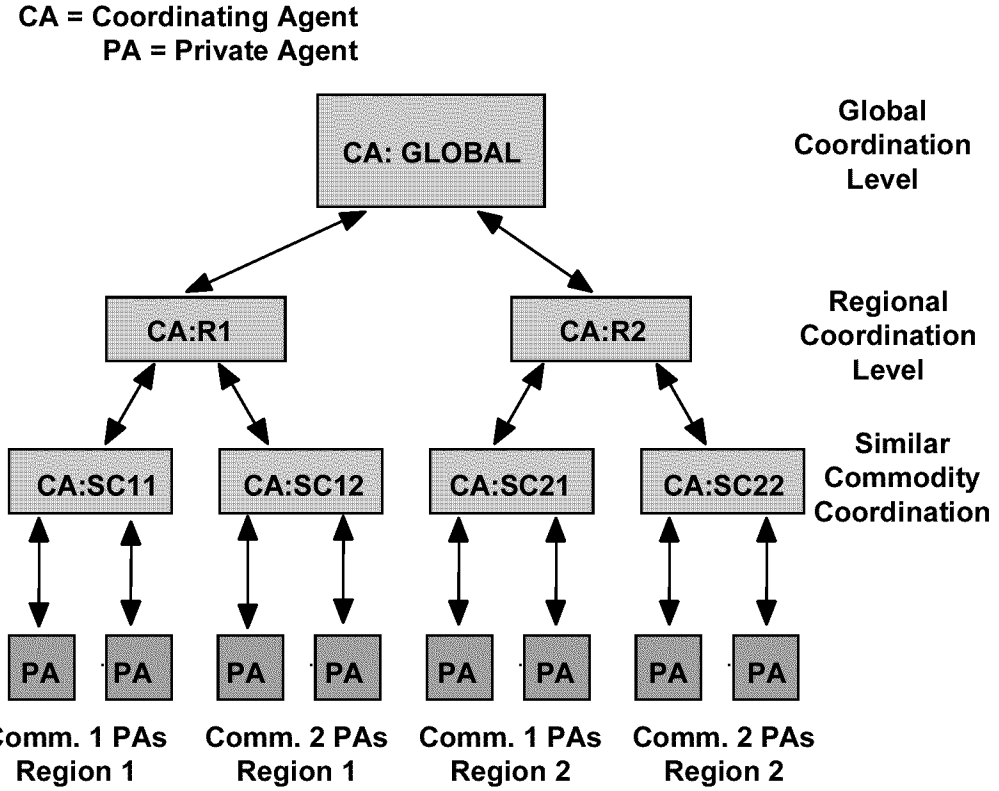

Figure 6.2. Hierarchical Decomposition and Coordination

In practice, PAs often construct abstractions of the PDNM that are used for planning and analysis, and are used as an interface to the regional coordinating authority for the industry. We call these *Private Agent Aggregated Network* (PAAN) models. A concrete example of the use of an aggregated PAAN can be found in the power industry. In the New England states, the New England Power Exchange (NEPEX) is an organization that coordinates several local power companies in the region (Boston Edison, Commonwealth Electric, Vermont Yankee…). Each local power company has provided to NEPEX an aggregated version of their private detailed power network. The aggregated power network includes generation nodes, consumer nodes and transmission lines exceeding 100KV. Thus, NEPEX has available a regional power network, for 100KV and above, connecting all New England utilities, including tie-lines to neighboring regions. Furthermore, data related to the 100KV network is provided by each power utility to NEPEX. Thus, at each instant of time, NEPEX has a high-fidelity model of power flows in the New England region. In case of severe emergencies, NEPEX issues directives to the local power companies and they (by prior agreement) implement NEPEX directives, by suitably transforming the NEPEX directives into detailed actuator signals to their physical power network.

Above the Coordinating Agents with responsibility for similar commodities in a region, we assume the existence of a Regional Coordinating Agent. This Regional CA is responsible for coordinating all the similar commodity CAs in the region. At the top of

this hierarchy, we find a global CA, typically a bureaucracy established by national political leadership. In practice, some levels of this hierarchy may of course be partially collapsed or missing entirely; see Figure 6.2..

6.4 Results

Reduced Order Modeling of Infrastructure Networks

The reduced-order modeling work on DARPA ENDSTATE centered on several experiments in a fictitious region where network models of electric power and POL distribution were developed using standard COTS packages, PSS/E and WinSliqFlow. Depicted in Figure 5.3 below is a map of the region of Troy, with the EP (dashed), POL (dotted), roads (red/gray) and rail (heavy black) networks pictured.

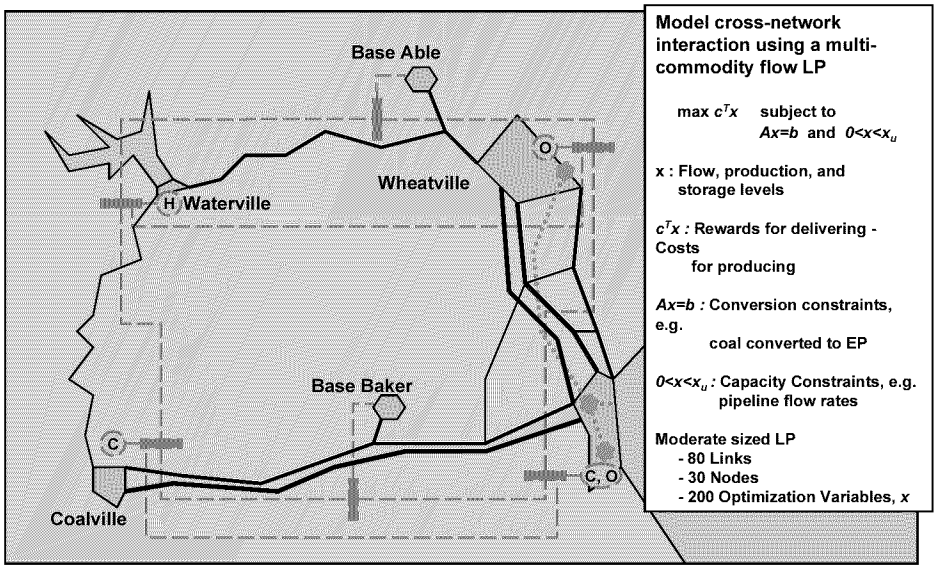

Figure 6.3. Trojan Networks

Key interdependencies among the networks include: Need to transport coal via rail from Coalville to Portville (lower right); need to transport POL products from Portville to Wheatville via pipeline; generation of EP from POL at Wheatville; generation of EP from coal at Coalville; generation of EP from either coal or POL at Portville; EP required to run pump stations along POL pipeline from Portville Wheatville. The rail line between Coalville and Portville is electrified and passable by diesel trains while the line between Portville and Wheatville is diesel-only.

Initial experiments were performed with a baseline composed of Linear Programming (LP) models. In fact, a first order approximation of cross-network coordination was

provided by the baseline LP methodology. By capturing constraints on commodity flow across model boundaries, and by providing objective functions that represent realistic penalties and rewards for consumption and storage priorities, the baseline includes embedded simple models of cross-network coordination. This is a possible approach to modeling what an adversary would or could do in response to system perturbations. A number of interesting results were obtained with the baseline models by introducing disruptions to production or supply shortages in specific locations in Troy and then following cascading effects through the composite model.

Work on ENDSTATE then proceeded to focus on the problem of building usable reduced-order models for two of the commodity networks in Troy and prototyping coordinating agents for the resulting PAAMs. The team first used Neural Net tools available from MathWorks to build a functional approximation of the behavior of the EP network in Troy. This worked well, providing a reasonable trade-off between the size of the hidden layer in the neural net and the resulting training time on one hand, and the accuracy of the outputs of the model on the other hand. However, when the same technique was applied to the POL distribution network in Troy, serious anomalies resulted. The pipeline between Portville and Wheatville was equipped with a number of pumping stations, powered by the EP network. Despite the fact that a very large number

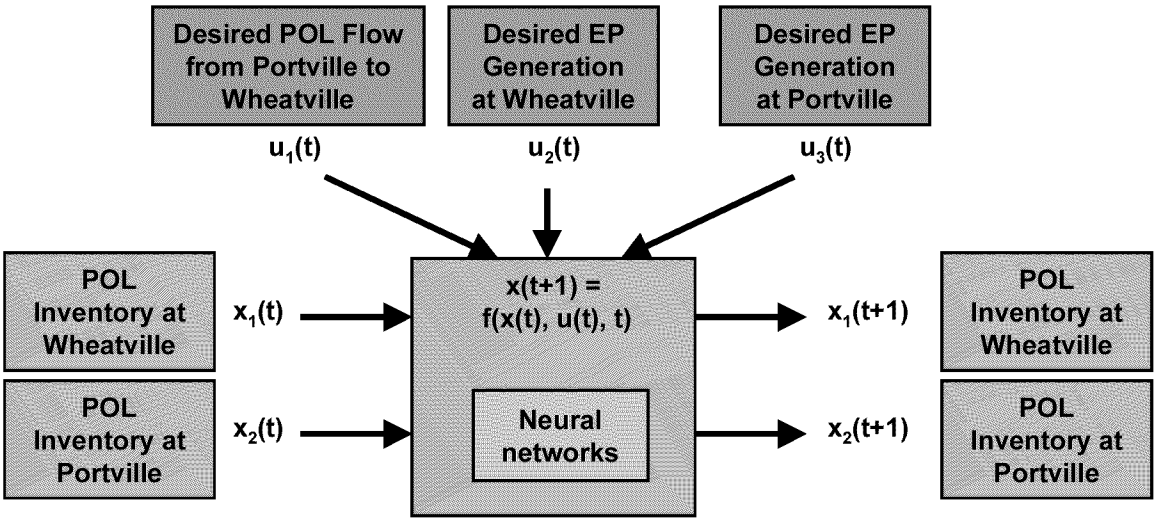

Figure 6.4. Cross-Network Coordination – The Dynamic Plant Model

of combinations of pump settings were possible, the equilibrium flows in the pipeline clustered around a surprisingly small set of discrete values. Instead of functional approximation, it turned out that a two-stage classification methodology with Neural Nets gave acceptable results, taking both accuracy and computational complexity into account.

Next, the we investigated the cross-network coordination problem. To be specific, two abstracted network models (PAAMs) were used as building blocks: EP and POL. Figure 6.4 above shows the experimental design in a graphical format. As described earlier, both were implemented as neural nets using MathWorks tools. As a prototype CA for these two networks, standard Dynamic Programming techniques were applied. When

viewed as a control problem, the state variables at each time t were the actual inventories of POL commodities at Wheatville and Portville. The control applied to the system was the desired POL flow in the pipeline and the target levels of EP generation at the two cities. In addition to the "true" state variables, secondary or derived state is summarized in the actual flow and generation and consumption levels.

We should emphasize that even this simple CA problem is intrinsically non-linear. The feedback in the state equation is one source of nonlinearity. Indeed the cross-network flows of supply and demand are non-linear in general as well, and include models of POL/EP conversion based on the heat equation.

Conclusions

Reduced order modeling via function approximation with neural nets works well for some infrastructure networks. In particular, the behavior of EP generation and transmission networks is amenable to this technique. Sigmoidal neural nets as approximations of (1) behavior that is based on time-series observations, and (2) target functions that are not monotonic tend to be inaccurate. However, the EP capacity/generation experimentation described above led to acceptable errors with significantly reduced computational complexity precisely because:

1. The non-linear behaviors of the system (e.g. its target functions) are not only monotone, they are continuously differentiable,
2. The underlying models being abstracted or mined are essentially static.
3. Function approximation does not work at all well for some infrastructure networks. Notably, the behavior of POL distribution is not well approximated using this approach. Care must be taken in choosing abstraction techniques, with assumptions verified by experiment.

The last experiments performed were a verification of the approximation errors embodied in the PAAM/CA methodology described above. Two major sources of possible errors were considered:

- Discretization assumptions in CA state and control space,
- Potential approximation errors in the neural net PAAMs.

Both types of errors were shown to be manageable.

7. Integrated COA Critiquing and Elaboration System (ICCES)

7.1 Problem

The ICCES Concept Experiment Program (CEP) Challenge Problem was conceived to develop, demonstrate and evaluate an integrated knowledge-based course of action critiquer and elaboration system (ICCES); see Figure 7.1. ICCES was designed to help battlefield commanders and staff rapidly (a) generate, critique, and wargame courses of

action (COAs), and (b) continuously revise COAs in response to changing situation estimates. The primary focus of the CEP was on developing new technology to support the operational requirements of the interim brigade.

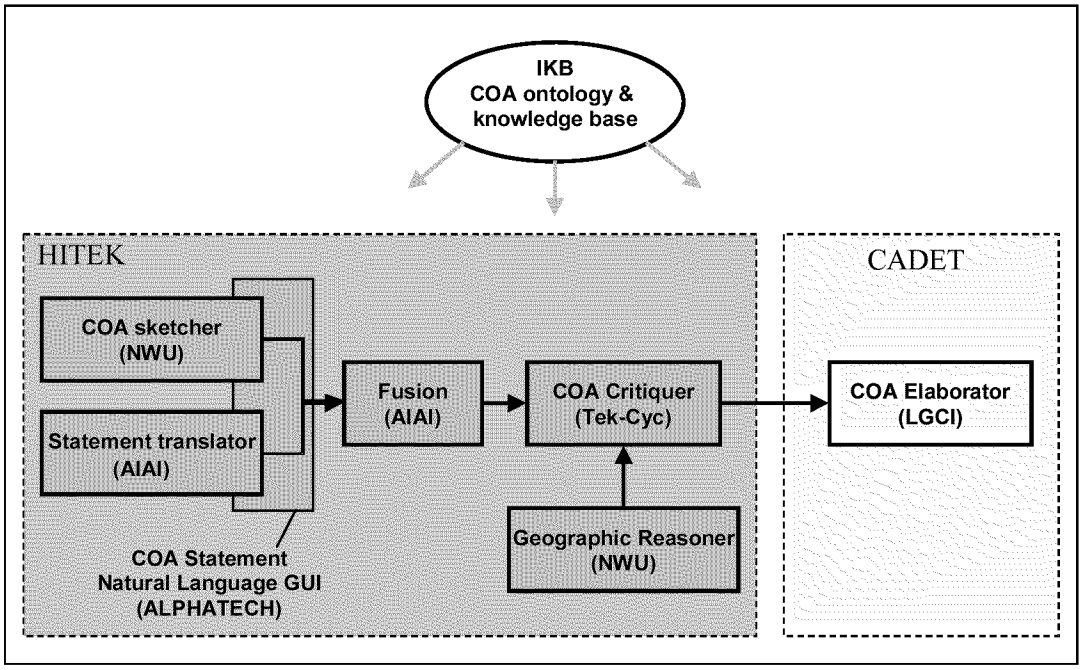

Figure 7.1. Integrated COA Critiquing and Elaboration System (ICCES)

Future battle command decision support systems will require the use of large-scale knowledge-based technologies to provide automated real-time feedback and predictive capabilities to aid decision-making. ICCES addressed this requirement by integrating the capabilities of HITEK, DARPA's High Performance Knowledge Bases COA critiquing prototype, and CADET, a CECOM SBIR prototype COA analysis tool.

The HITEK critiquing tool consisting of the following components:

(1) a knowledge-based sketch input tool developed by Northwestern University (NWU) that accepts a combination of speech and freehand sketching,

(2) a COA statement input tool developed by the AI Applications Institute (AIAI) and Teknowledge,

(3) a graphic user interface (GUI) tool developed by ALPHATECH, Inc.,

(4) a "fusion module" developed by AIAI that combines the output of the COA statement input tool and the sketch input tool,

(5) a geographic reasoner developed by NWU, and

(6) a course of action critiquing associate developed by Teknowledge and Cycorp that can systematically analyze and suggest refinements to a COA.

The components employ a shared integrated knowledge base (IKB) to represent knowledge and communicate information between components.

CADET is a prototype decision support system that helps a user develop detailed tactical plans, including a detailed Synchronization Matrix, unit tasks, activities and routes, estimates of battle losses, weapons system and personnel attrition, fuel and ammunition consumption, and enemy reactions and counteractions. CADET takes as input information on the friendly and enemy forces, a description of the battlefield, elements of a COA sketch consisting of a rough scheme of maneuver, and a list of specified tasks. CADET then produces, under user control, a detailed plan, appropriate for dissemination to subordinates either via a synchronization matrix or a 5-paragraph Operations Order. CADET was developed by Logica Carnegie Group's predecessor Carnegie Group, Inc. with SBIR Phase I and II funding under the sponsorship of US Army CECOM and with guidance from the Battle Command Battle Laboratory - Leavenworth and the 4th Infantry Division at Ft. Hood.

7.2 Objective

The objective of the ICCES CEP was to integrate the HITEK and CADET systems in order to provide an initial end-to-end automated capability for generating and analyzing multiple COAs in detail, in less time, and with reduced level of effort. ICCES was envisioned as a prototype knowledge-based decision support system to facilitate rapid commander-centric development and analysis of COAs for brigade and division level operations, supporting continuous enroute mission planning and C2-on-the-move, and enabling commanders and staffs to rapidly generate, analyze, wargame, and continuously revise COAs using a prototype automated battle planning decision support system.

ICCES offered a potentially revolutionary capability for conducting commander and battle staff training, by facilitating rapid generation of planning products, and by providing real time feedback and recommendations on feasibility, acceptability, and suitability. The system would enable the user to rapidly change initial conditions to generate an action/reaction response cycle with which to develop and evaluate commander and staff proficiency.

7.3 Approach

The approach for the ICCES CEP involved three basic steps: developing scenarios and a challenge problem specification (CP spec), building the ICCES system by integrating the HITEK and CADET systems, and conducting experiments to evaluate the results. These are described in more detail in the following paragraphs.

Development of Scenarios and Challenge Problem Specification

The CP Spec for the ICCES CEP was developed as an extension to the Course of Action Critiquing Challenge Problem Specification developed for Year 2 of the HPKB program. The ICCES CEP CP Spec defined inputs, processing, and outputs for the ICCES CEP.

The inputs provided are in many respects similar to those provided in the HPKB Year 2 COA Analysis Challenge Problem: the format of products of mission analysis (PMA), COA statements and sketches, and relative combat power (RCP) analyses are almost unchanged from HPKB. However, several new items were also required for integration of HITEK with CADET: a mobility corridor graph, and a list of specified and implied tasks and associated temporal constraints among them.

The ICCES CEP CP spec contained three brigade-level scenarios:

- A *training scenario* taken from the HPKB COA CP Final Evaluation (known there as Item 5, Scenario 4),
- Two related *test scenarios* based on Prairie Warrior '00.

For each scenario, the following items were provided:

- Products of Mission Analysis (PMA), as for the HPKB COA CP,
- Initial Blue COAs,
- Critiques of the initial Blue COAs, and model answers,
- Repaired Blue COAs, i.e. revised versions of the initial Blue COAs that address the critiques,
- List of specified and implied tasks in the repaired Blue COAs,
- Temporal constraints between those tasks.

Note that this version of the CEP CP spec provides only an initial Blue COA and not a repaired Blue COA, so the list of specified and implied tasks and associated temporal constraints provided are for the initial Blue COA.

Building ICCES

Figure 7.1 depicts the components of ICCES, their origin and how they fit together. As part of the process of building the ICCES system, ALPHATECH's natural language GUI for inputting COA statements was integrated as a part of NWU's NuSketch tool. The resulting COA Statement window in NuSketch allows the user to input natural language sentences describing:

- Close Battle
- Deep Battle
- Rear Battle
- Reserve
- Security

- Fires

Each of these categories has its own tab in the COA Statement window (see Figure 7.2). The user may freely switch between tabs without losing previous work in other tabs.

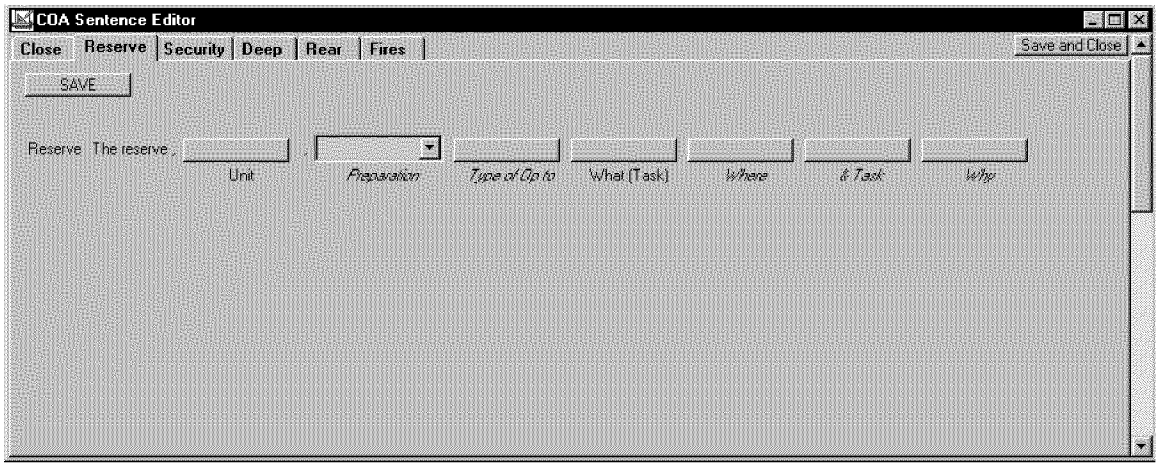

Figure 7.2. COA Statement window, showing template for Reserve statements.

Rather than allowing free text input, which is very difficult for computers to process and understand, the Statement window provides a *template* that ensures the sentences will be structured according to a simple grammar. This template gives users some degree of freedom in expressing tasks and purposes without compromising the computer's ability to understand them.

Figure 7. Depicts the overall functionality of ICCES, focusing on the input products the CP Spec provided. The box labeled "Critique COAs" represents the capability of HITEK. The boxes with dashed borders are input items provided with the CP Spec scenarios. The (oval shaped) processing steps outside the "Critique COAs" box represent additional work that needed to be done to integrate HITEK with CADET; these various steps contributed to the creation of a mobility corridor graph and a list of tasks in the repaired COA with temporal constraints. COA refinement was then carried out by CADET.

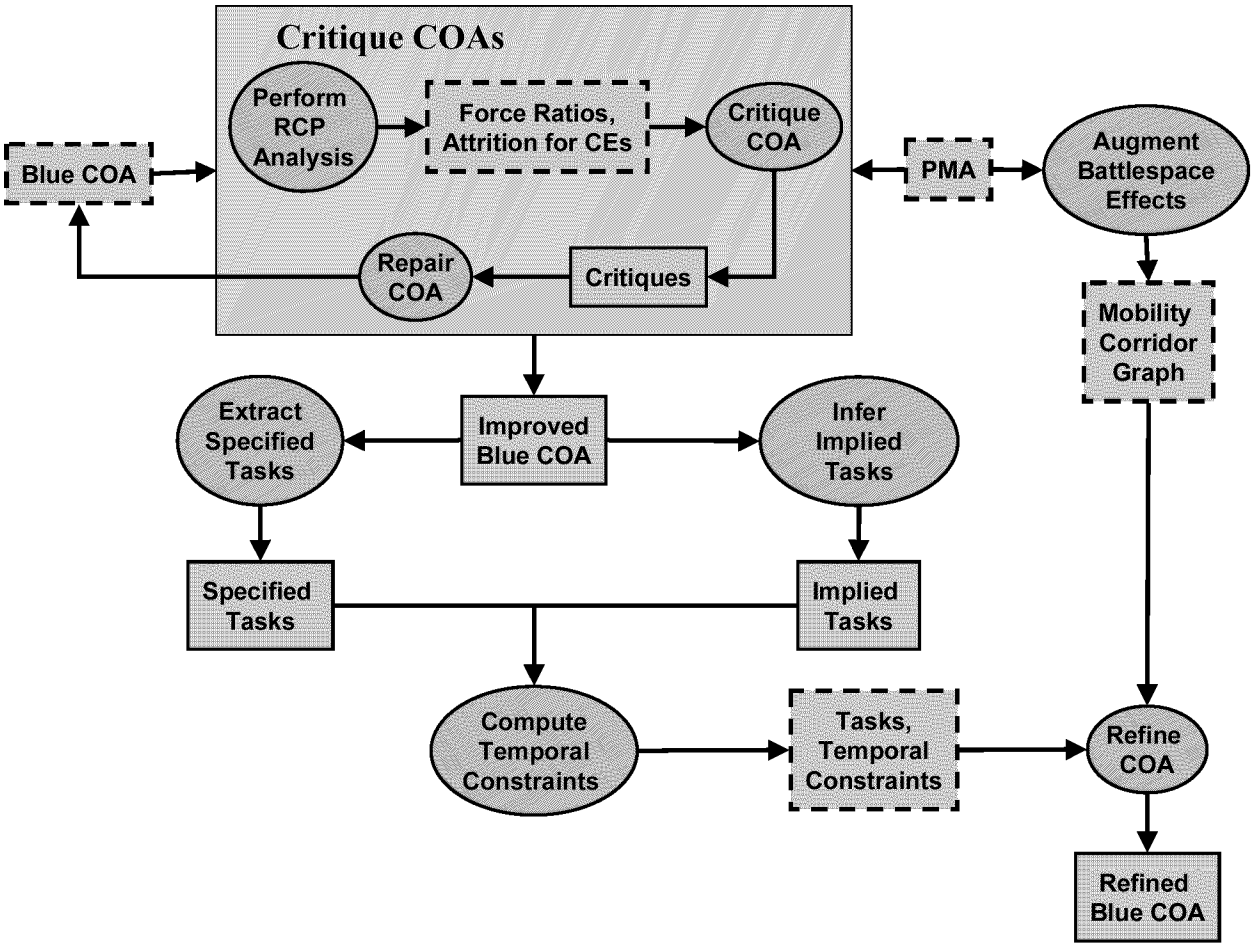

Figure 7.3. ICCES information requirements

Conducting Experiments

A series of battle lab experiments were conducted to evaluate ICCES, drawing on Command and General Staff College students as subject matter experts. The experiments assessed whether a prototype knowledge-based decision support system can rapidly develop and analyze COAs for brigade and division level operations, and rapidly revise them in response to an evolving situation. These experiments attempted to quantify the benefits of using the prototype along several dimensions, by comparing the performance of groups of subjects using the prototype to other groups not using the prototype. Dimensions evaluated included speed of COA development, analysis, and revision, and final plan quality. The experiments also attempted to identify areas of missing knowledge and functionality, and ways in which the system's interactions with the user could be further improved to best support en route mission planning and commander-centric C2-on-the-move.

Measures included the amount of time required to generate courses of action and an operations order, and assessments of the suitability, feasibility, acceptability, completeness, strengths, and weaknesses of these plans.

Training and evaluation scenarios were developed using the methodology utilized during the HPKB COA Analysis Challenge Problem. A subject matter expert (SME), an active duty US Army officer who supported HPKB in a similar role, developed the initial scenarios, which were then interactively refined to conform to the input specifications of ICCES, with the SME providing support and quality control. The training scenario was a scenario developed for the final evaluation of the HITEK system during the HPKB program. The evaluation scenarios were based on the Prairie Warrior '00 exercise; two brigade level scenarios were derived from the division level scenario for PW '00.

7.4 Results

The ICCES CEP provided a valuable window into the workings of an automated plan generation system, including both positive and negative potential effects. In the structured experiment, the automated system did not succeed in speeding up the process; in fact, the group not using the automated prototype system finished well before the ICCES system group (mainly attributable to mechanical problems of using the software, due to insufficient training time). Another criticism of the automated system is that the staff loses the "training" benefit of performing the analysis by hand – a process by which the commander and staff come to understand the situation sufficiently well that when events go differently than planned (which is generally the case), they are well-prepared to adapt because of their thorough understanding of the battlefield situation.

On the other hand, it was generally agreed that the system was doctrinally sound, and the fact that the system required a user to specify a purpose for every action was considered a very positive approach (because the purpose is usually more important than the task). The plan developed with the ICCES system, capitalizing on its automated support, was evaluated to be a significantly better-quality plan (after addressing the mechanical problems mentioned) than that developed by the group not using the ICCES system. The system was described by various members of the test group as being somewhere between a 20% and an 80% solution.

8. BioSurveillance Seedling Study

8.1 Problem

A terrorist who desires to affect a large number of people with a biowarfare agent will choose an agent whose symptoms do not immediately alert officials to the attack, and thus induce many casualties. For example, the initial symptoms of anthrax are very similar to those of influenza, so a terrorist may choose an airborne anthrax attack in November or December, when these symptoms would not immediately alert officials. The BioSurveillance CP study seeks to examine potential technologies to detect such previously undetected covert attacks, by looking for patterns of disease symptoms in a population that are inconsistent with standard disease propagation, while there is still time to reverse the effects of the agent.

There are no effective systems in place today to provide an early detection of such an attack. In fact, the initial casualties of an anthrax attack may actually be misdiagnosed as some other respiratory illness, and the true cause of death may not be detected until large numbers of people have been killed, and it is well past the time when an antidote will be effective. The challenge problem is to investigate methods to detect such an attack early enough to prevent widespread casualties. Of course, the most effective way to detect an anthrax attack is to develop and deploy sensors that can detect anthrax, thereby providing warning as soon as the anthrax is released. A premise of this work is that the initial attack has gone undetected, perhaps because such sensors are only deployed at large gatherings, and can not blanket an entire city, or because the biowarfare pathogen was engineered to avoid such detection.

Complicating the ability to correlate the early reports of the attacks is the potentially distributed nature of the reporting. For example, consider an attack on the Arlington, VA business district during mid-day. The people that will be affected are primarily those that work in this area—but who live in many different parts of Virginia. During the first few days after the attack, these people will start feeling ill, and will stay home. Eventually, they will go to their doctors or to their hospitals, most likely in the places close to where they live, not where they work. Thus, the early warnings will be distributed geographically, and may not become obvious without the ability to link people together. Thus, a significant challenge is to identify the patterns of activity by providing links between individuals or groups of people.

There are several clear areas where we can improve the current situation:

- *Focus on the early symptoms of the disease relative to normal levels of diseases*. There are clear patterns of the upper respiratory symptoms that fluctuate throughout the year, especially as the yearly influenza epidemic spreads throughout the community. Epidemiologists currently focus on static models of how diseases are passed between people, and then look for links between these people. We need to generalize this concept both to look at

communities of people, and to look at the dynamics of the propagation in real time.

- *Group people not only by location, but also by other socioeconomic factors.* We will never be able to model every individual in a population, and therefore we should look at groups of people, based on factors such as geography. However, as anyone with young children is aware, diseases propagate very rapidly in schools, and from the schools to the families and workplaces. If possible, we want to consider factors such as family structure (e.g., whether there are school-aged kids), age, profession, etc. The challenge is in choosing factors for which supporting data are available.
- *Use dynamic models of disease propagation.* Disease levels are by no means static, and there are clear trends as the disease propagates through the community. In order to detect anomalies, we need models of the normal disease levels. These models must be dynamic, capturing the propagation of disease throughout the population. The models must capture the uncertain nature of disease propagation, and need to be both calibrated to each year's version of influenza, and also constantly compared to the actual levels of diseases.
- *Use early indicators of illness to look for anomalies.* The normal level of influenza can be estimated based upon the number of deaths caused by flu. More effective for our problem is obtaining reports from doctors on the numbers of people who come in to clinics with flu-like symptoms. Even earlier indicators include the number of bottles of flu medicine or other groceries purchased, such as apple juice or chicken soup. These data are actually being collected, and could potentially be put to use for monitoring the health of the general population.
- *Allow for varying levels of privacy.* Health data is very personal, and we must ensure that there is no loss in privacy in setting up such a health-monitoring system. When a real crisis exists, however, it is widely accepted that private information may need to be accessed. Therefore, there should be varying levels of privacy protection.
- *Use this system solely as an early-detection system.* Correlating health data can provide circumstantial evidence of an anomaly in the spread of contagious diseases, but it will be very unlikely that such an approach can say with any definiteness that an anthrax attack has occurred. So we must allow for this system to be an early detection system, which helps direct other collection assets. For example, when a potential anomaly is detected, doctors and hospitals can be notified to be on the lookout for specific symptoms and to run specific tests. Furthermore, particular sensors could be deployed around a particular location to test for the presence of a pathogen. Other means can also be used to gain further evidence that a biowarfare attack has occurred.

Clearly there is a great need for such a system, and there is a great potential to improve our ability to detect such attacks early enough to prevent widespread casualties. In this

task, we focus on developing the technologies to model and predict the spread of diseases. There are a large number of additional health-care and coordinated response issues that are not addressed, as they do not relate to detecting the patterns of activity in large amounts of health data that we focus on here.

8.2 Objective

The focus of this task was to study the feasibility of developing early detection systems, and to develop the appropriate dynamic models of disease propagation. Such models, often called transmission or network models in the field of epidemiology, provide a statistical representation of how a disease transmits between different groups of people. These groups are separated by geography, age, or other socio-economic factors, and within a group we assume people interact homogeneously (that is, there is an equal likelihood of any person interacting with any other person—an assumption which can only be correct for very small groups). The model includes the probability of people both within a group and between groups coming into contact, and the probability that a communicable disease propagates from one person to the next. Disease data, such as latency and incubation periods, are also included in the model.

For this BioSurveillance Seedling study, we will assume that we have health indicator data, which could include grocery and pharmacy purchases, doctor or hospital reports, and school absenteeism. While not traditionally considered health indicators, these factors do provide noisy measurements of the propagation of diseases such as influenza within a population. For example, grocery purchases of fluids vary according to the time of year and sales at stores, but are also affected by people trying to alleviate flu symptoms. An increase of these purchases relative to a nominal threshold is an indication of flu-like symptoms within the local population. Thus, our health data monitoring must incorporate the varying levels of "normal" activity.

We will use these health data and the dynamical models to estimate the number of people who have communicable diseases, and to detect statistical anomalies. There are two steps in this process. Given an accurate dynamical model, we will use nonlinear estimation theory to estimate the number of sick people based upon the health data. But first, we need to estimate the parameters of the dynamical model, using data from a period of time in which we are certain there is no biological attack, and apply nonlinear parameter estimation techniques.

Our estimator is then used for a biological warfare detector. An agent such as airborne anthrax is not communicable, and would show a very different disease pattern than influenza. Thus, we use our model to detect statistical anomalies relative to the expected pattern of activity in a normal flu season.

Initially, our efforts will focus on feasibility studies, to demonstrate that this modeling framework can detect biowarfare attacks, and to determine the magnitude of such attacks that can be reliably detected. We will use an optimistic viewpoint: we will use simulated data, and assume that the models are correctly calibrated and validated, and measure our ability to detect biowarfare attacks. This will provide the "best-case" performance, and provide us with 1) an assessment if this detection approach is feasible even in the best

case, and 2) performance goals that we will seek to achieve when building the real biowarfare detector.

8.3 Approach

The overall approach can be seen in Figure 8.1. The health of the general population is affected both by contagious diseases, which propagate around the communities, and biowarfare attacks. We do not observe the number of people who are sick directly, but must rely on several different indicators. Potential indicators, as shown in the figure, include

- doctors reports—these are the reports that are currently filled out by physicians for general record-keeping and for reimbursement from insurance companies;
- pharmacy purchases—these include sales of prescription drugs (that ultimately could be tied to individuals if the situation warrants), as well as more general nonprescription drugs;
- grocery purchases—sales of apple juice, chicken soup, and other clear liquids increase during the flu season, and they, along with other grocery purchases, can give insight into the state of the flu epidemic when compared to normal levels;
- school absenteeism—diseases often spread rapidly around schools, and absenteeism may be one of the first indicators of a sudden increase in illness; and
- other factors that will become apparent as our work continues.

All of these factors can be affected by means other than illnesses. For example school absenteeism may be affected by religious holidays, exams, and community events. Grocery purchases can be affected by discounts or holidays, and other factors. So each of these specific detectors needs to be normalized relative to the external sources that affect them. Stated another way, we consider each of these factors to be a type of sensor of disease levels, and need to normalize them in order to identify the signal (infectious disease behavior) from the background clutter.

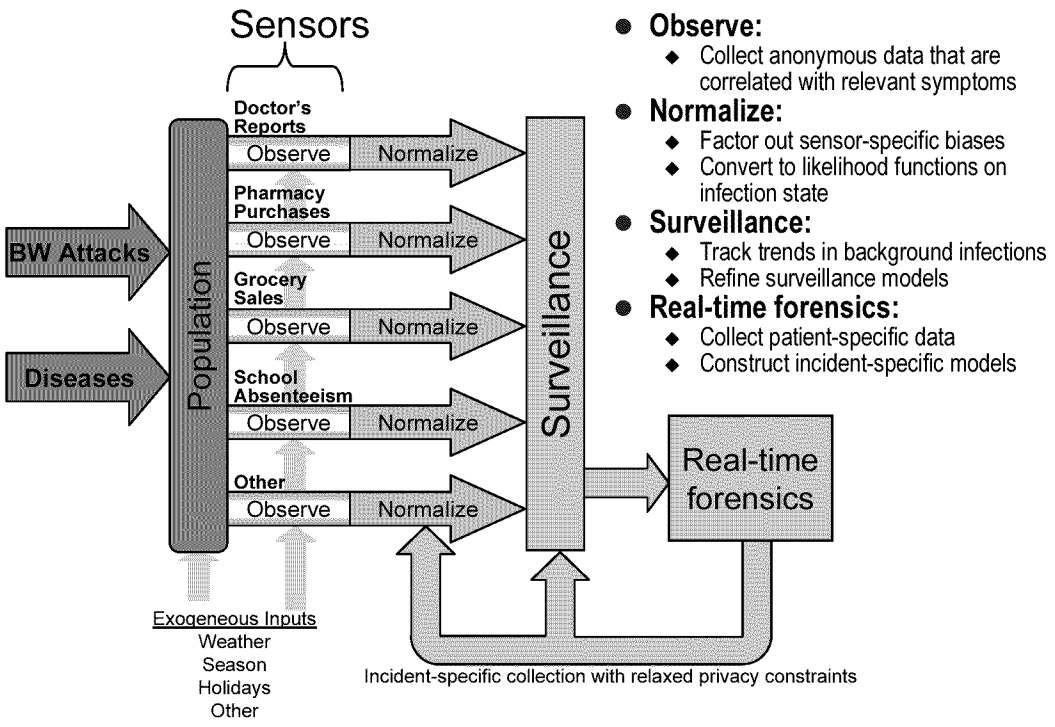

Figure 8.1. System Concept: Surveillance Triggers Specialized Processing.

Each of these normalized signals is then fed into an overall surveillance box. It is this surveillance box that is the focus for the challenge problem. This box contains the real-time epidemiology model that captures the behavior of infectious diseases in a probabilistic model. Using the observations from each of the normalized detectors, we apply nonlinear estimation theory to update our current estimate of the level of disease within the population. We also use this process to detect anomalies that could possibly be explained by a biowarfare attack.

The output of the surveillance box is an anomaly for further investigation. The real-time forensics process directs more detailed data to be obtained, either by directing health-care workers, or deploying specialized sensors to specific areas. Furthermore, a sufficiently suspicious activity may warrant relaxing the privacy constraints imposed upon our models, providing more details as to the interactions between people and the spread of disease.

Let us consider in more detail our approach to the surveillance box. This process contains a model of the spread of disease among a population. We will use a dynamical model of the spread of diseases, that contain different possible states (e.g., susceptible, contagious, infected, recovered, etc.) for people. A simple version of such a model is shown in Figure 8.2. Initially, we begin with a population of people who are susceptible to disease. With some probability, they will become infected by some other person, and during the incubation period they are contagious but show no symptoms. This state will last anywhere from one to four days (as determined by a probability distribution), until symptoms begin to show. They will remain in this state anywhere from three to seven

days (again, given by a probability distribution) until they recover. A small percentage of people die each year from influenza, and so we include that state as well.

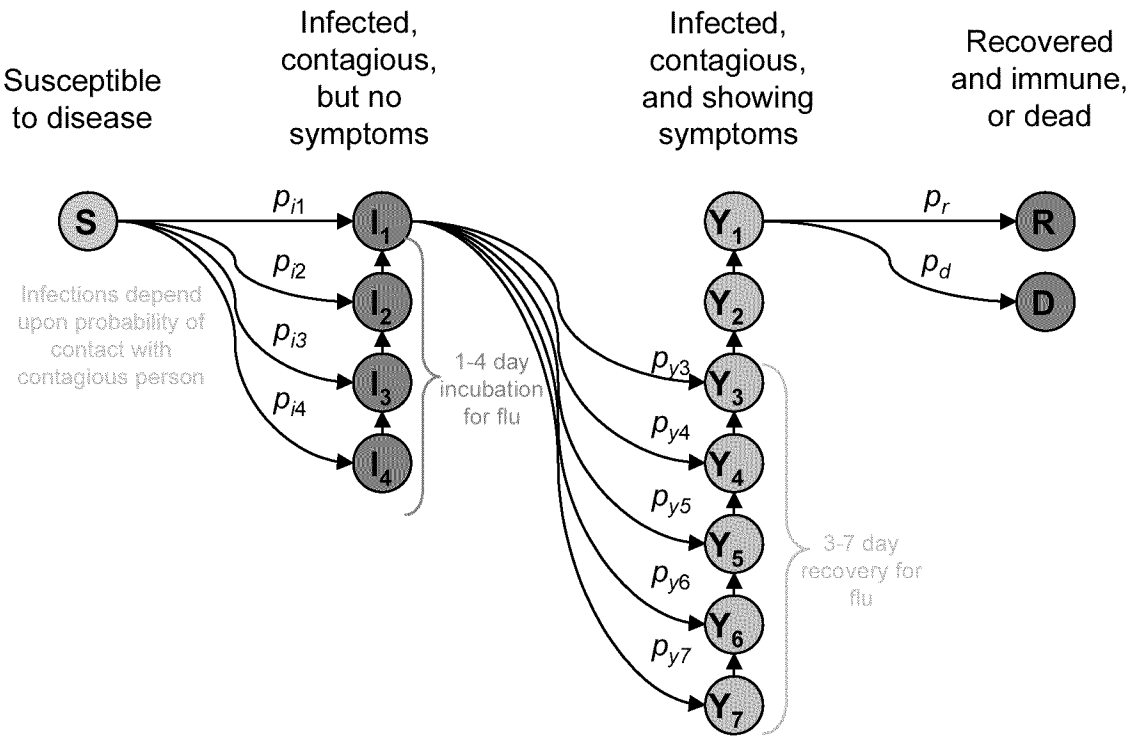

Figure 8.2. Disease Spread is described by a Markov Chain.

Clearly there are many ways that we can make these models more complicated, depending upon the desired fidelity of the representation. For example, our current assumption is that once recovered, a person is immune from further illness. In reality, there are varied levels of susceptibility to different diseases, and more sophisticated models could capture this. Of course, we need to first understand if such fidelity is required in our models.

An important focus of our model is the disaggregation of the population into different groups of people. As shown in Figure 8.3, there are many levels of aggregation in which we can model the population. At the coarsest scale, we could lump the entire population into a single group. However, a key assumption of these groups is that there is homogenous mixing; that is, there is an equal likelihood that any two people in the group will interact, and thus provide an opportunity for the spread of disease. Clearly this is not true. The other extreme is to model each individual as a group. The trouble here, of course, is that we could not possibly model the detailed relationships between individuals without personal knowledge on each individual. Instead we need to consider small demographic clusters of people. In our initial studies, we consider each small demographic cluster to be homogeneous, and probabilistically model the interactions between people both within and between groups.

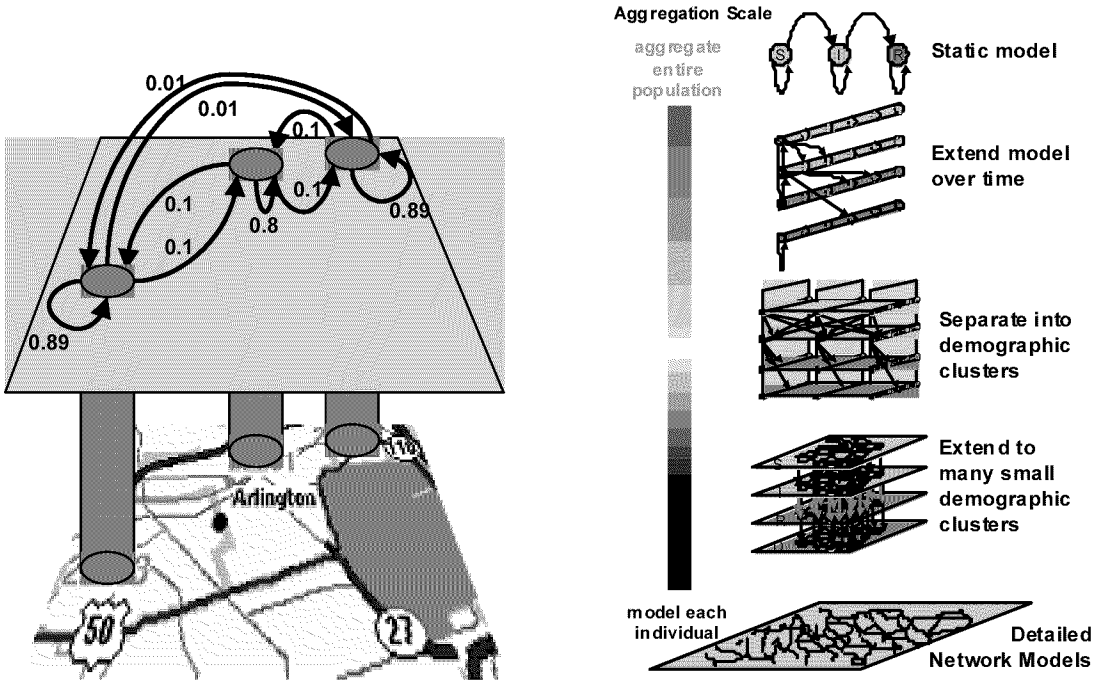

Figure 8.3. Transmission models indicate the likelihood of coming into contact with a person in a different location. They can be built at multiple resolutions to conform to data availability and demographic structure.

8.4 Results

Our results include both a summary of the model we derived, as well as typical performance results.

Both the dynamical and spatially varying models are based upon discrete-time within-household infection models, known as Reed-Frost or Greenwood models. The likelihood of a single susceptible person coming into contact with a contagious person from cluster *i* is:

$$1-\left(1-p_{trans}k^{i,j}/P^{i}\right)^{C^{i}_{t-1}} \approx p_{trans}k^{i,j}C^{i}_{t-1}/P^{i} \quad \text{for small} \quad p_{trans}k^{i,j}/P^{i}$$

where P^i is the population in cluster *i*, C^i_t is the number of contagious people in cluster *i* at time *t*, $k^{i,j}$ is the likelihood of a person in cluster *j* interacting with someone in cluster *i* (see Figure 8.3), and p_{trans} is the probability of transmission of the disease when two people come into contact (which may be variable, but is constant here for simplicity).

The total number of new infections is given by a Bernoulli trial for each of the S^j_t susceptible people, leading to a binomial random variable n^j_t with mean and variance given by

$$E\{n_t^j\} = p_{trans} \sum_i k^{i,j} C_{t-1}^i S_{t-1}^j / P^i ,$$

$$Var\{n_t^j\} = p_{trans} \sum_i k^{i,j} C_{t-1}^i / P^i \left(1 - p_{trans} \sum_i k^{i,j} C_{t-1}^i / P^i\right) S_{t-1}^j$$

The dynamical model uses these equations to represent the number of new infections, and uses a probability model, as shown in Figure 8.2, to model the changes in state from infected, to contagious, to recovered or dead. We also include the probability that we observe an infected person through one of the sensors in the surveillance system, as a function of the number of days since infection.

For the purpose of this feasibility study, we use this model to simulate the normal flu season, as well as a biowarfare attack (using different parameters for each model), and also optimistically assume that the estimator uses the correct model parameters. Of course, these parameters need to be estimated from the data in a deployed system, but this approach allows us to calculate the best-case performance. Figure 8.4 shows typical results from our simulator, including both the background influenza infections, and the additional infections from the BW attack. Note that our goal is to detect these very subtle changes in the number of infections.

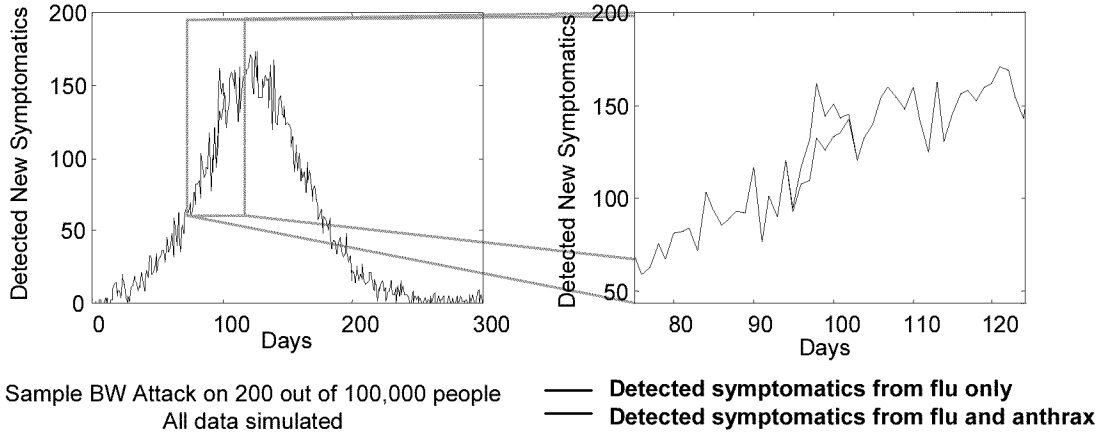

Figure 8.4. Attack signatures are small compared with clutter.

Our detector is based upon the Extended Kalman Filter. For this filter, we approximate all Bernoulli random variables with Gaussian random variables with the same mean and variance. Thus, the number of newly infected people in cluster *j* is given by

$$n_t^j = p_{trans} \sum_i \beta^{i,j} C_{t-1}^i S_{t-1}^j + \sqrt{p_{trans} \sum_i \beta^{i,j} C_{t-1}^i \left(1 - p_{trans} \sum_i \beta^{i,j} C_{t-1}^i\right) S_{t-1}^j} \, w_t^j$$

$$w_t^j \sim N(0,1)$$

We have therefore rewritten the noise as the white Gaussian term w_t^j, uncorrelated with other noise terms. Linearizing this equation around the current estimates of our state variables, (denoted with a '^') and the mean of w_t^j (which is 0), we can write

$$n_t^j = p_{trans}\sum_i \beta^{i,j}\hat{C}_{t-1}^i S_{t-1}^j + p_{trans}\sum_i \beta^{i,j} C_{t-1}^i \hat{S}_{t-1}^j - p_{trans}\sum_i \beta^{i,j}\hat{C}_{t-1}^i \hat{S}_{t-1}^j$$
$$+\sqrt{p_{trans}\sum_i \beta^{i,j}\hat{C}_{t-1}^i\left(1 - p_{trans}\sum_i \beta^{i,j}\hat{C}_{t-1}^i\right)\hat{S}_{t-1}^j}\, w_t^j$$

This is now linear in the state variables C_{t-1}^j and S_{t-1}^j, and depends upon the estimates of these state variables, obtained from the previous time step.

The key idea in using the Extended Kalman Filter for detection of statistical anomalies is shown in Figure 8.5. By dynamically tracking the course of the infections (i.e., the epidemiological state), we estimate not only the number of people who are infected, but also a set of *covariance statistics* that indicate our uncertainty in the estimate, both spatial and temporal. A statistic of interest, called the *innovations*, indicates the variation in our data that is unexplained by models and past data. A statistical anomaly is one in which the innovations exceed a threshold, such as a 5-sigma value.

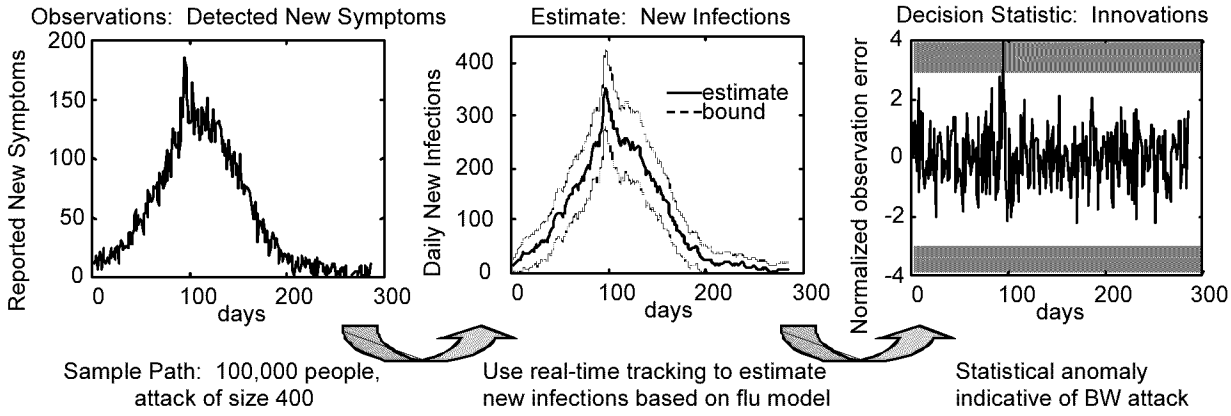

Figure 8.5. We dynamically track the number of naturally occurring infections to improve BW attack detection.

The foundation for our approach is optimal estimation of the state of (nonlinear, large scale) Markov processes. Estimators derived from this foundation operate in four steps: 1) the epidemiological state estimate, and variance, from one day get predicted forward, using our models of disease progression and transmission, to an estimate of infection levels at the next day; 2) expected observables are derived from this prediction; 3) the expected observables are compared with actual observables to compute innovations; and 4) if innovations are small, the difference is used to update the epidemiological state estimate. Large innovations indicate a statistical anomaly that should be investigated further.

The success of a statistical anomaly detection algorithm is best measured by examining the tradeoffs between the probability of detection versus the probability of false alarm. Lowering our detection threshold (i.e., the size of the innovations that we declare indicates an anomaly) increases the probability of detection, but also increases the false alarm rate. A graph of such tradeoffs, such as shown in Figure 8.6, is commonly called the Receiver Operating Characteristic (ROC) curve. Choosing the parameters of our

detector (including the detection threshold) allows us to operate at different points along this curve. Note that, for a specified false alarm rate, the algorithms in this study have a higher probability of detecting the attack by waiting longer—but these are exactly the delays that may prove fatal.

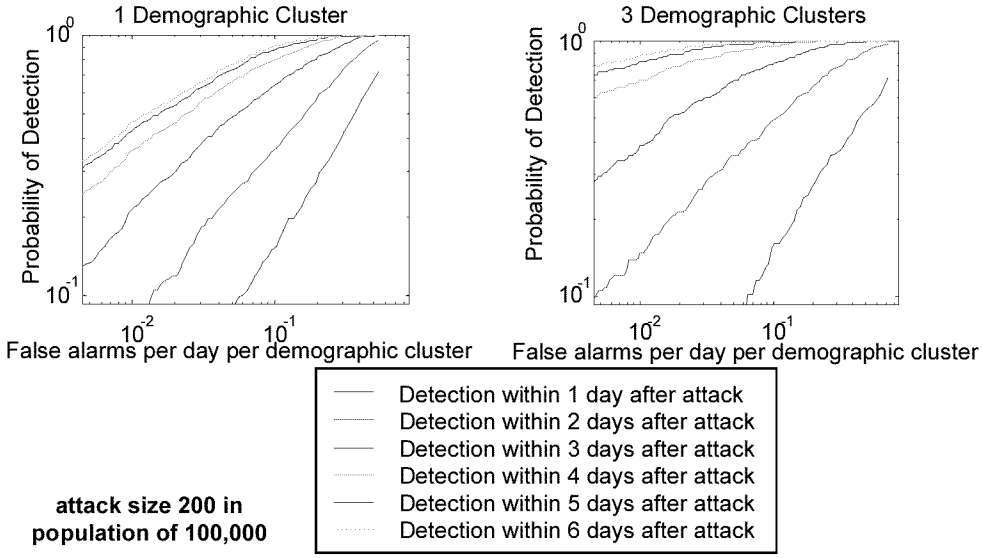

Figure 8.6. Detection performance trades detection probability with false alarms. Increasing the number of clusters improves performance.

Our results clearly show that such a detection system is feasible, in the best case. We can detect relatively small attacks (200 out of 100,000 people), and the performance greatly increases as we can separate these people into multiple demographic clusters. Of course, these results are optimistic, based on the assumption of an accurate model structure and correctly estimated parameters. Further study should examine these assumptions in more detail. Specific areas for further work are the following:

- Develop nonlinear parameter estimation techniques in order to calibrate our models to specific disease data. The parameters of our model should be continually updated, in order to conform to the current years influenza characteristics. For example, an influenza virus very similar to the previous year's virus will not spread as quickly, as many people will already be immune to it. Of course, we want to estimate the parameters on sets of data for which we are confident do not contain the effects of a biowarfare agent, and then use the resulting model to look for statistical anomalies as new data arrive.
- Study the effect of both incorrect parameter estimates, and also incorrect model structures (which contain, for example, the wrong number of states or potential transitions between states in Figure 8.2 that are not modeled).
- Perform sensitivity studies to determine the critical parameters for accurate detection performance. For example, the homogeneous mixing assumption of

the cluster models is incorrect, but we do not yet know the significance of this approximation.

- Use real data to measure the performance of the system. Since we certainly hope any real data does not contain a real anthrax attack, we should inject the effects of such an attack into any data (real or simulated) that we are using for analysis.

9. SHIELD (Super Human Information Extraction and Link Discovery)

9.1 Problem Definition

The goal of the EELD (Evidence Extraction and Link Discovery) program is to develop technology to detect, identify, classify, and generate hypotheses about a variety of Asymmetric Threats, e.g., terrorist threats, proliferation (nuclear, biological, chemical, and missile), and transnational crime. The SHIELD task is a seedling effort to identify current and emerging technologies that can help address this problem, identify new technologies needed to meet the needs of Asymmetric Threat analysis, and develop recommendations for future investments. Technologies explored included information and evidence extraction, link discovery and analysis, and probabilistic link models.

Asymmetric Threat

One by one, they obtained photo identification. Two by two, they rented mailboxes and bought airplane tickets. Three by three, they took new apartments and visited bars. Five by five, they killed innocents by the thousands… Seen in isolation, the activities have the mundane feel of daily American life… but when the activities are viewed in the context of Sept. 11, patterns emerge of men living quietly, separately and anonymously, then suddenly acting in concert, collectively perusing a horrible goal on a tight timetable… Despite the occasional missteps, terrorism authorities say, the Boston group fit the profile of a "sleeper cell."

– "How 10 hijackers hid in plain sight", *The Boston Globe*, 9/23/2001, p. A1.

For months and even years, the hints were everywhere -- phone intercepts in India, warnings from a man arrested in Germany, even public bragging by Osama bin Laden. Watch lists maintained by the CIA, federal trial transcripts in New York, tips to German police from an Iranian in Hamburg; offhand remarks by already captured terrorists. All for naught, it turns out -- each warning seemed pretty interesting at the time, but was not specific enough, or credible enough to warrant full-scale alarm. In the end, few of the warnings were tied into all the others to see whether there was a discernible mosaic here, something you could grab onto…

> "When we start looking at damage assessments," said Frank Cilluffo, director of the terrorism task force at the Center for Strategic and International Studies in Washington, "we will see all these individual strands of information. In retrospect, we will see how the dots can be connected."

– "In retrospect, wisps of many clues hinted at attack. But each on its own not enough to stitch into credible fabric." *San Francisco Chronicle*, 9/24/2001, p. A3.

As the tragic events of 11 September 2001 amply demonstrate, preventing future terrorist attacks and other forms of asymmetric threat, is a national security priority. Asymmetric threat encompasses a wide range of activities involving non-state, transnational entities such as Osama bin Laden's Al-Qaeda organization, as well as third and fourth world nations who are actively engaged in attacking or developing the capability to attack US

interests in non-traditional (i.e., force-on-force) ways. These activities include terrorist attacks, proliferation (missile, nuclear, biological, and chemical), and transnational crime (organized crime, including money laundering, stock and banking fraud, and drug trafficking). Looking just at terrorist groups, it is estimated that there are 50 to 60 active terrorists groups, with a total of 100,000 core members. There are an estimated additional 300-500 radical, criminal, or fringe groups, with a membership of 1,000,000, from which the active terrorist organizations can draw or which can turn into active terrorist groups themselves. Members of these groups frequently try to hide their activities by blending into the surrounding environment, disguising their movements and financial transactions, and by operating decentralized operations in which the individual participants in a particular action may not even know each other. The challenge is to detect the presence of these groups and their activities prior to the commission of a terrorist or other asymmetric threat event (e.g., proliferation) or to identify and prosecute less catastrophic events (e.g., money laundering and other forms of fraud). Meeting this challenge requires developing a suite of technologies that can detect, track, and identify asymmetric threats by fusing information from ultra large-scale, multiple, heterogeneous data sources.

9.2 Objectives

The goal of the SHIELD CP was to identify current and emerging technologies that can help address this problem, identify new technologies needed to meet the needs of Asymmetric Threat analysis, and develop recommendations for future investments. The CP program proceeded in two phases. In the first phase, the various seedling contractors conducted a variety of feasibility studies and assessments, including limited technology development, designed to address these goals. In the second phase, the seedling contractors focused on conducting more specific experiments using some of the more promising technologies demonstrated at the Science Fair.

Phase One Objectives

During the first phase of the program, ALPHATECH's objectives included the following:

- Develop an unclassified challenge problem that could be used to: a) refine the system concept; b) demonstrate the capabilities of current emerging technologies to perform limited EELD; c) identify the challenges and additional technology investments required to develop a full EELD capability.
- Provide additional support to the program including a) coordinating the science fair; b) developing and maintaining a program web site; c) maintaining a program mailing list; and d) generating technology summaries and briefing charts.

Phase Two Objectives

During the second phase of the program, ALPHATECH's objectives included the following:

- Develop a classified challenge problem for the purpose of conducting an experiment using the iterative classification software developed by David Jensen (University of Massachusetts).
- Coordinate the experiment with DARPA, Schafer (SETA contractor), Dr. Jensen, the CIA's Advanced Technology Programs office, and the CIA's Weapons Intelligence, Non-Proliferation, and Arms Control (WINPAC) office.

9.3 Approach

Phase One Approach

Due to security restrictions associated with Asymmetric Threat data, the first step was to identify an analog to the Asymmetric Threat problem. The analog needed to have characteristics in common with the Asymmetric Threat problem (the presence of detectable behavioral patterns; relationships between and among people and organizations; events of interest; a wide variety of data sources; etc.), but for which there is some available ground truth. In conjunction with DARPA, we selected U.S. business relationships as a reasonable analog. The initial version of the challenge problem focussed on the banking and chemical companies. The final version included thirteen industries.

The approach to developing the challenge problem involved the following four steps:

1. Identify and acquire data on U.S. businesses. Data sources included Securities and Exchange Commission (SEC) data for corporate reporting, Lexis-Nexis for news and press releases, and industry-specific media sources (e.g., American Banker). ALPHATECH purchased data from these sources and used it to develop the challenge problems.
2. Develop an ontology for describing the types of data and relations of interest.
3. Extract appropriate data sets from the data sources (free text for the information extraction developers and structured data for the pattern learning developers).
4. Distribute the ontology and data to the technology developers via the SHIELD web site.

Phase Two Approach

In the second phase of the program, we focused on coordinating an experiment whose goal is to apply the iterative classification software called Proximity, developed by David Jensen (UMass-Amherst) to classified data provided by the CIA. For security reasons, the type of data and the goal of the classification cannot be discussed in this report. The approach to this has included the following steps:

1. Perform knowledge acquisition with members of the CIA's Weapons Intelligence, Non-Proliferation, and Arms Control (WINPAC) center to: a) specify the data, and b) define the goal of the classification.

2. Develop a surrogate problem description to facilitate unclassified interactions with UMass.

3. Install Proximity software and conduct detailed timing and sizing studies on a standalone, secure system at ALPHATECH, using synthetic data to determine the current processing bounds and to pre-test the software. This includes working with UMass to harden their software.

4. Develop preprocessing software tools to perform data cleaning.

5. Define an experimental procedure.

6. Conduct the experiment and report the results.

Under the seedling funding provided through the HPKB contract, we completed steps 1, 2, and parts of 3, 4, and 5. We are continuing work on the seedling experiment under funding through another contract vehicle.

9.4 Results

Phase One Results

Phase One Challenge Problem Selection

There were four criteria for selecting the challenge problem in the first phase of the SHIELD effort. These were:

1. The CP must be unclassified to accommodate participation by academic researchers.

2. Data must be available and there must be some reasonable assumption that ground truth exists and could be acquired.

3. The data should be qualitatively similar to real world asymmetric threat data; the desired qualitative similarities included: dynamic affiliations of humans to organizations, complex and evolving relations among organizations, high-value activities requiring several months of preparation, partially revealed information in text format, initially hidden authoritative process and signature models, and the addition of value by the detection of specific people at specific locations.

4. The data should be mathematically similar to real world asymmetric threat data; the desired qualitative similarities included: having an object-relational state space, discrete-event dynamics, and the possibility of ambiguous and non-unique assignment of observations to state elements.

The behavior of American corporate culture, especially with respect to mergers and acquisitions, was identified as the challenge problem surrogate. This general area meets the first two criteria: namely that a business based challenge problem is unclassified and provides a large amount of data, for which there is a reasonable expectation of ground truth (i.e., SEC data). The activity patterns between business practice and asymmetric threat are analogous. Table 9.1 illustrates some of the qualitative similarities between the

surrogate and real problems, addressing the third criteria. Business problems also address the fourth criteria, as listed above. (Note, the business data may not have the same statistical characteristics as the classified data; this was not something we considered in the seedling effort, as we were not able to perform any validation.)

ALPHATECH purchased data from several sources: the Securities and Exchange Commission, Lexis-Nexis, and American Banker. This data is a combination of structured and unstructured (i.e., raw text) information. Preparing the raw text data for use was relatively straightforward. Transforming the SEC Primark raw structured data into an ontology-based relational database format required developing a series of Perl scripts to extract, format, and clean the structured data.

Table 9.1. Qualitative Comparison of Business and Asymmetric Threat Domains

Business Activities	Asymmetric Threat
Organizational: • Mergers • Spin-offs • Joint ventures • Cross-industry directorates	Organizational: • Alliances • Splinter groups • Coalitions • Common funding sources, suppliers, specialists
Personnel: • Recruiting • Promotions • Resignations (voluntary & otherwise) • Transfers	Personnel: • Recruiting • Role changes • Resignations (voluntary & otherwise) • Transfer of allegiance
Financial: • New stock issue • New bond placement • International transactions	Financial: • New funding source • New front business • Money laundering
Targeting: • Market research • Sales calls • Proposal preparation • Contract award	Targeting: • Surveillance • Reconnaissance • Weapon construction • Attack

Phase One Challenge Problem Ontology

In conjunction with the information extraction and pattern learning contractors, we developed an ontology for describing relationships within and between companies. Four versions of the ontology were released, culminating in version 0.3.2. The final ontology is comprised of three sub-ontologies: company relations, financials – balance sheets, and financials – stock performance. Each of the components in the ontology provides the capability to completely describe internal and external company relations, both

personnel-related and asset-related. The 0.3.2 version of the ontology uses a relational database format.

There are two main components of the ontology: objects and links. The objects include: person, organization, owner, contractor, SIC, address, and source. The links identify which objects are associated with each other, how they are associated, and the date the information was published. Examples of links include employee, role, company status, and company owner.

The ontology and data sets are available on the SHIELD web site (see below).

Challenge Problem Results

The challenge problem materials were provided to the EELD seedling contractors via the SHIELD web site (see below) prior to the Science Fair (held in April 2000).

System Concept Refinement

Figure 9.1 shows a simplified view of the EELD system concept that evolved over the course of the seedling effort. (Note, more detailed versions have been developed by DARPA since work was completed under the HPKB contract. This version is included here for completeness.) First, there is a data stream that is constantly being processed; this data represents traditional and maybe other sources of intel data. Information extraction techniques are used to extract relational fragments (person A was seen at training camp Y in the fall of 1998). Over time, the information extraction system will learn and refine its search.

The extracted fragments are then stored in a link data repository; as shown in the chart, new fragments may need to be correlated with information that is already contained in the repository (i.e., ambiguities may need to be resolved, etc.). As the repository is populated with data, the analysts can review it and provide feedback in the form of patterns of interest. The system will classify the data and present hypotheses regarding potential instances of the patterns of interest to the analyst. In addition, the system will seek additional information to confirm or disconfirm individual instances.

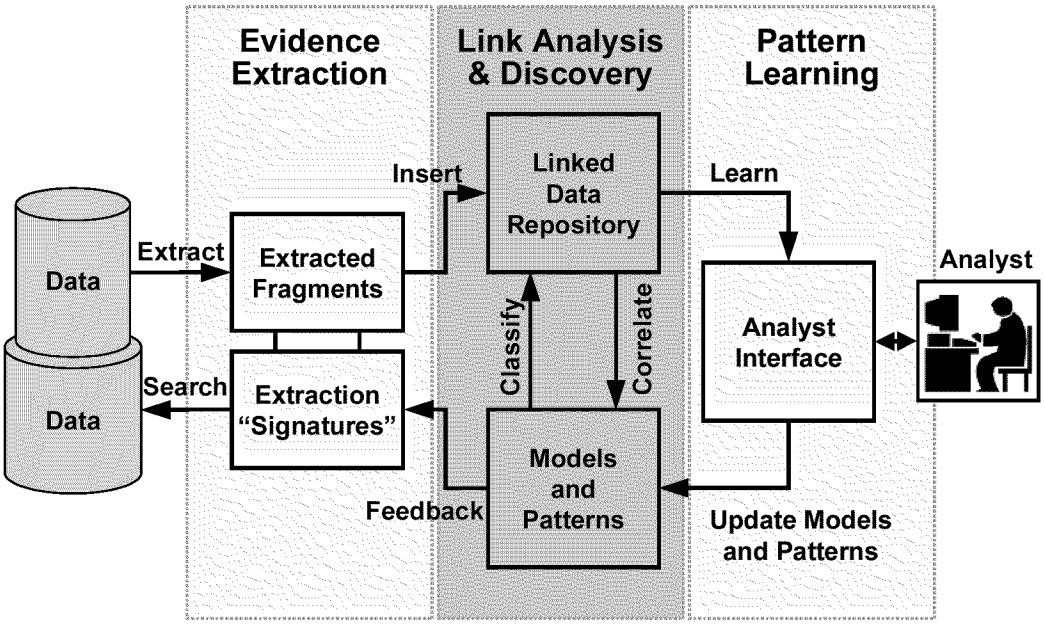

Figure 9.1. Simplified Overview of the EELD System Concept.

Phase Two Challenge Problem

As noted above, phase two challenge problem development was partially completed under funding provided through the HPKB program. Work is on-going to complete the phase two challenge problem experiment through another funding vehicle. In this report, we document our progress under the HPKB program.

Surrogate Version of the Phase Two Challenge Problem

Due to classification restrictions, we developed a surrogate problem to facilitate discussions with Dr. Jensen at UMass-Amherst, whose software is being used in the experiment

Proximity User Testing, and Timing and Sizing Experiments

Under the HPKB program, we conducted extensive user testing and preliminary timing and sizing experiments. Throughout the user testing process, we reported any problems to UMass, and they, in turn, provided software updates. After each update, we reinstalled the software and conducted additional user testing. The early user testing focused on basic Proximity functionality. Later testing focused on exercising the queries we expect to use in the actual experiment. On-going timing and sizing experiments (being conducted under a separate contract) will enable us to determine the bounds on the problem size as a function of the graph size and characteristics.

Pre-processing Software

We are developing three types of pre-processing software: data analysis software to calculate graph theoretic statistics, data reduction software to filter out expected low incidence regions, and data transformation software to convert the input data into

Proximity format. Preliminary software design was conducted under the HPKB contract; implementation is on-going under another contract vehicle.

Experiment Design

One of the challenges we face in conducting this experiment is that we will not be able to start with a fully labeled training set. Therefore, we need to develop a surrogate for a labeled training set. As a result, the experiment will proceed in two stages, the first of which has two major parts, as described below.

Experiment 1A

The goal of Experiment 1A is to develop labels for all objects in the data set, so that the fully labeled data set can be used for training the Proximity classifier in Experiment 2. We will achieve this by extracting objects of interest based on a pre-determined pattern. In this case, the pattern is a path from a Producer region to a consumer region. Once the "interesting" objects are extracted, we will generate the "interesting" paths that they lie on. We will then prioritize the "interesting" objects by the length of path on which they lie, the types of regions connected by the paths, and the number of paths from node to regions of interest. These prioritized lists will be passed to WINPAC analysts who will be asked to review the top 100 or so objects to assess whether or not they would have found them interesting.

Experiment 1B

The goal of Experiment 1B is to perform a trial run of statistical classification before Experiment 2. We will use an analyst watch list to confirm a subset of the interesting objects. Use a sample of uninteresting objects (not appearing on watch list) as a mini-fully labeled training set. We will then train the Proximity classifier on the small set, generate labels for the remainder of the data set, and compare the labels to the labels in 1A.

Experiment 2

The goal of Experiment 2 is to use the fully labeled training set that results from Experiment 1A (together with analyst corrections to the interesting node set) to train the Proximity classifier, and then use the classifier to predict the labels on the objects of a test data set. The output of Experiment 2 will be passed to the WINTEL analysts for review.

SHIELD Web Site & Mailing List

ALPHATECH established a program web site and mailing list. These were used to distribute and archive data, experimental results, program briefings, and program announcements, as well as information regarding the asymmetric threat.

The SHIELD web site is maintained at ALPHATECH in a protected project area. The SHIELD web can be accessed at:

www.alphatech.com/protected/shield

with the following username and password:

Username:	shield
Password:	Superhuman

10. TBM REASONER

10.1 Problem

Theater Ballistic Missile (TBM) units present a number of significant challenges to the warfighter. They are highly mobile and easily concealed, but at the same time they have the potential to affect theater-wide operations. Key TBM vehicles include the Transporter Erector Launcher (TEL), missile resupply transporter, crane, compressor truck, neutralizer truck, and other support vehicles. During field operations, these TBM vehicles can be found at the Forward Operating Base (FOB), Forward Operating Location (FOL), Transload Location (TL), hide site, and launch site.

In a typical TBM attack cycle (see Figure 10.1), an unloaded TEL will emerge from an FOB or a hide site and will rendezvous with a missile resupply transporter, crane and other support vehicles at a TL, where a missile will be fueled, armed and loaded onto the TEL. Upon completion, the loaded TEL will transition to a pre-launch hide site. In the final attach stage, the loaded TEL will transition to a launch site and prepare to launch its missile. Minutes after the launch, the TEL will quickly go to a post-launch hide site where it can begin a new attach cycle.

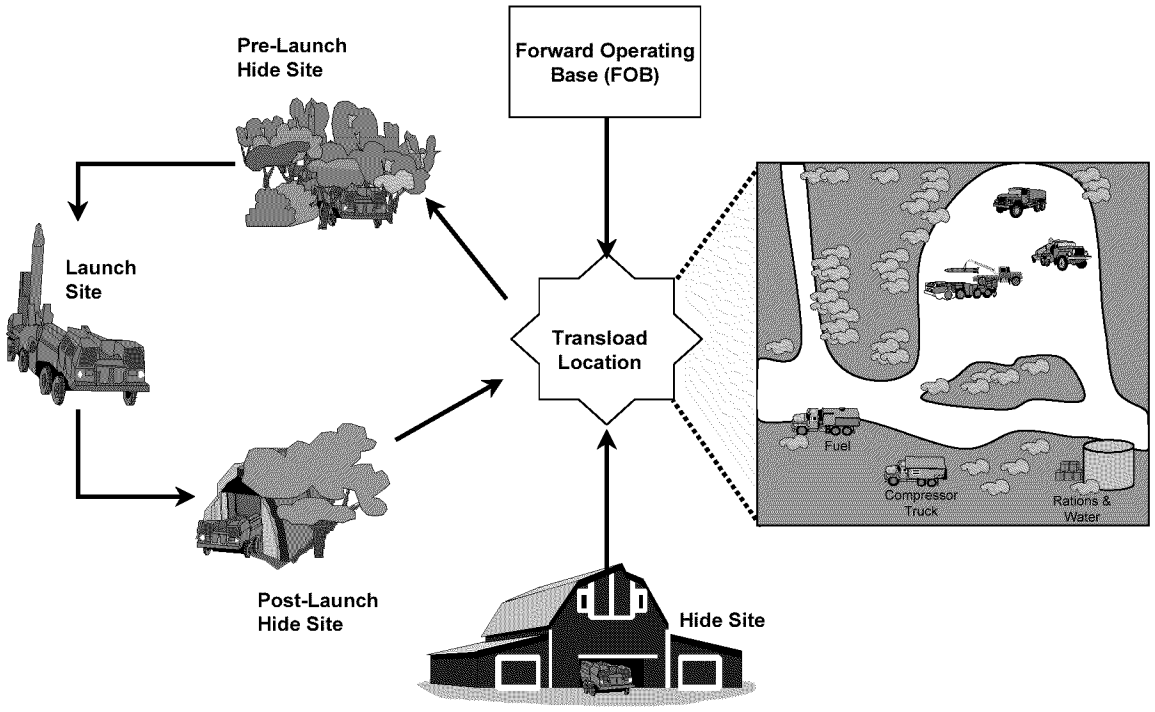

Figure 10.1. Typical TBM firing unit attach cycle.

A recently published multi-service document describing tactics, techniques, and procedures for Theater Missile Defense Intelligence Preparation of the Battlespace details current best practices that intelligence analysts use to rapidly locate and track TCTs associated with TBM operations. The document specifies a four-step IPB process to aid identification of these TCTs:

1. *Define the battlespace environment*. This step collects initial intelligence about the battlespace such as terrain, weather, logistical infrastructure, and demographics.

2. *Defining the battlespace effects*. This step evaluates the constraints the environment imposes on friendly and enemy operations.

3. *Evaluate the adversary*. This step determines how the adversary TBM force normally organizes for combat and conducts operations. For well-known adversaries, this process can rely on historical data and well-developed threat models. For less well-known adversaries, intelligence analysts may need to develop new intelligence databases and threat models.

4. *Determine adversary COAs*. This last step integrates the results of the previous steps into concrete threat hypotheses that form the basis for TCT detection and prosecution.

Each step of the TMD IPB process leads to the generation of specific IPB products that can aid the analyst in identifying TBM vehicles in the battlespace. Key IPB products include the following:

1. *The area of operations (AO) for TBM activities*. This product is developed during step 1 of the IPB process outlined above.

2. *Avenues of approach/mobility corridors for TBM vehicles*. This product is developed during step 2 of the IPB process outlined above.

3. *Prior knowledge regarding the locations of TBM sites*. This knowledge is derived by integrating two IPB products: specific prior intelligence (recorded during step 3 of IPB), and an "area limitations" analysis that determines terrain constraints on possible site locations (determined during step 2).

4. *Spatial deployment templates and branches-and-sequels templates*. These products are developed during step 3 of IPB, and encode doctrinal patterns of enemy TBM employment. Spatial deployment templates specify constraints on typical distances and travel times between TBM sites. Branches-and-sequels templates are generic state transition models encoding the launch and resupply cycles, abstract away from the details of the specific terrain. These include estimates of the amount of time taken to perform key TBM activities such as transload and launch, and they also include constraints on movement rates for TBM vehicles (which are themselves derived in part from the physical performance limitations of these vehicles).

Once an analyst has completed the IPB process, the COAs developed therein currently guide a manual analysis of available sensor data, including very large amounts of GMTI track data, to identify patterns of activity consistent with the COAs. Analysts seek patterns of behavior at NAIs consistent with TBM operations, including appropriate vehicle movements between hypothesized TBM sites, and vehicle operations within TBM sites. However, due to the large amount of available data, analysts quickly become overloaded. Studies have shown that 80% of the available sensor data never gets examined by the analysts; moreover, much of the data that is analyzed may be examined too late to inform target prosecution.

10.2 Objectives

The objective of the effort was to develop an advanced, knowledge-based decision aid that helps analysts rapidly identify TCTs associated with TBM units, and demonstrate it during the Joint Expeditionary Force eXperiment (JEFX) 2000 at Hurlburt Field, FL. The TBM Reasoner provides an automated pattern-matching algorithm that rapidly matches tracks derived from GMTI data against the detailed geo-registered COAs to identify potential targets and estimate their activities. In so doing, the system improves tracker performance by stitching together track segments corresponding to single TBM vehicles engaged in TBM operations.

The decision aid (a) exploits GMTI data, (b) directly supports emerging Theater Missile Defense (TMD) Information Preparation of the Battlespace (IPB) doctrine, and (c) addresses currently unmet requirements of the emerging suite of Air Force TCT decision aids. The resulting capability identifies TBM objects (e.g. TELs, resupply vehicles) and activity states (e.g. loaded TEL en route to launch site) in near real time.

10.3 Approach

The prototype system developed, the JEFX Testbed, consists of two components: (a) the *Traffic Generator* and (b) the *TBM Reasoner*. The Traffic Generator is a scripting tool that allows the user to generate TEL vehicle movements (tracks) in accordance with doctrinal TBM behavior templates, and merge these TEL tracks with canned civilian vehicle tracks. The TBM Reasoner accepts as input the vehicle tracks generated by the traffic generator and:

- Detects and classifies TBM vehicles in track data based upon the battlespace environment, battlespace effects, doctrinal behavior templates, and current intelligence estimates
- Uses observations of movements of suspected TBM vehicles to reduce uncertainty in estimated location of suspected TBM sites
- Provides a map-based display that highlights tracks corresponding to identified TEL vehicles, depicts inferred TBM vehicle activities, and displays locational uncertainty of TBM sites

Traffic Generator

As previously mentioned, the Traffic Generator allows the user to generate TEL vehicle movements (tracks) according to doctrinal behavior templates. The TEL doctrinal behavior templates used in our application was derived from the *Theater Missile Defense Intelligence Preparation of the Battlespace TTP* and is presented in Figure 10.2.

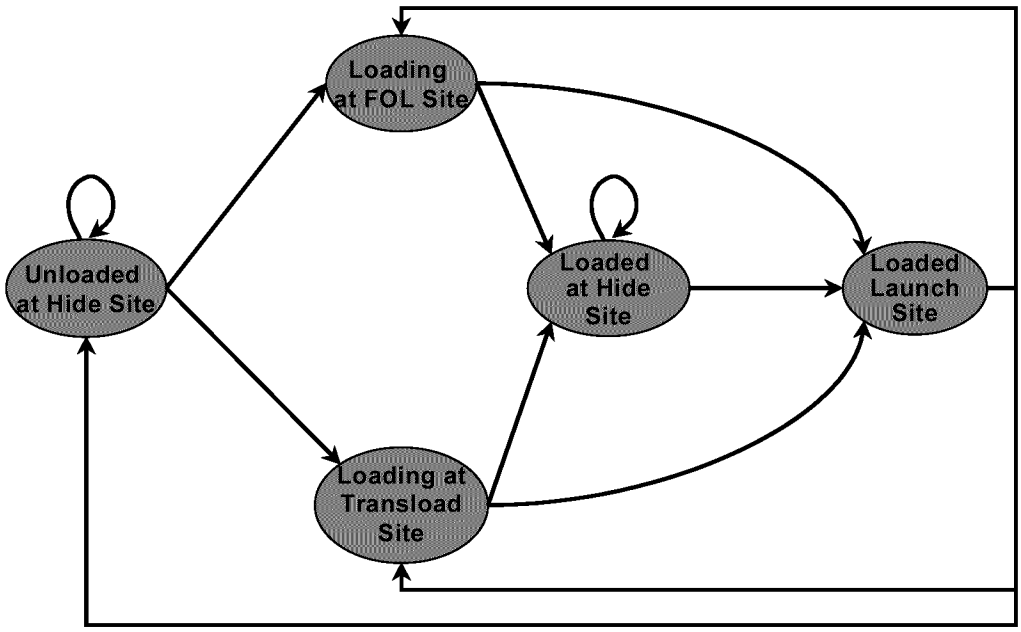

Figure 10.2. TEL doctrinal behavior template.

In Figure 10.2, we depict the states that a TEL can be in (oval objects) and the allowed transitions between states (arrows). For example, a TEL that is *Unloaded at Hide Site* can transition to one of the following states: *Loading at FOL Site, Loading at Transload Site,* or transition to its current state: *Unloaded at Hide Site.*

This template is embedded within the Traffic Generator to constrain the users to develop tracks consistent with doctrinal TEL behavior. For example, if a user has scripted a movement that results in a TEL at the state *Unloaded at Hide Site,* the Traffic Generator will allow the user to script TEL movements that will transition the TEL state into one of the three allowable states, namely *Loading at FOL Site, Loading at Transload Site* or *Unloaded at Hide Site.*

TBM Resoner

The approach adopted for the TBM Reasoner (see Figure 10.3) takes as input key products of the IPB process, including the battlespace environment, battlespace effects, prior intelligence regarding key enemy sites and activities, enemy combat power estimates, enemy doctrinal behavior templates, and hypothesized enemy courses of action (COAs). Knowledge representations of these products are automatically compiled by the *Model Compilation Tool* into detailed probabilistic state transition models encoded as Hidden Markov Models (HMMs). Within the *Pattern Recognition Engine*, these HMMs are then automatically matched against GMTI track data to identify suspicious patterns of vehicle activity conforming to postulated enemy COAs, enabling timely ISR or attack missions.

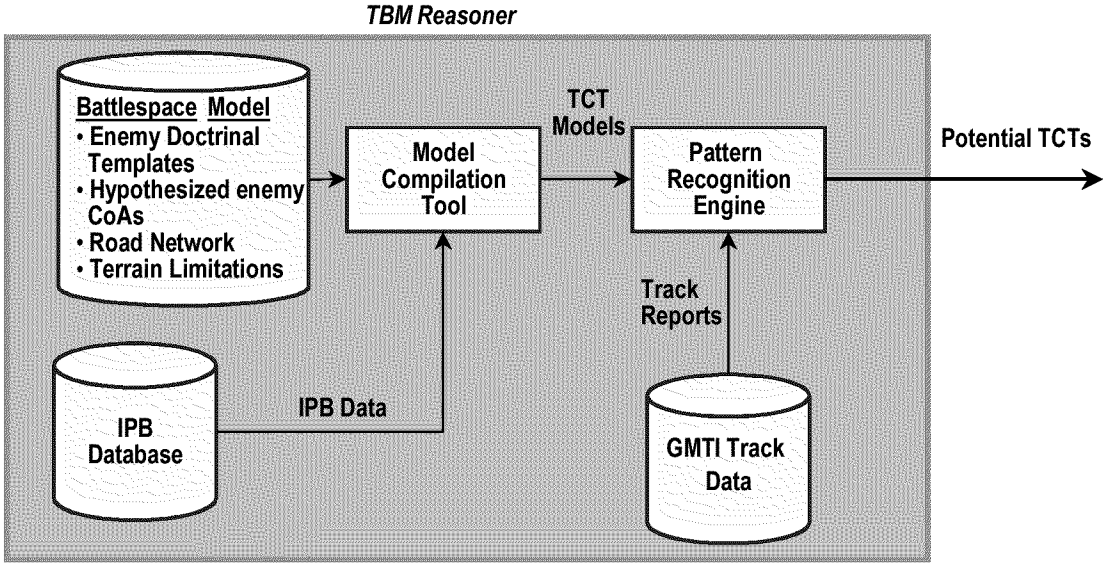

Figure 10.3. The TBM Reasoner encodes doctrinal templates, enemy COAs, and other IPB products to infer physical patterns of enemy activity from GMTI track data.

10.4 Results

The demonstration scenario presented during JEFX 2000 was based on the region of Kosovo. As part of the TBM Reasoner initiative, a knowledge-base consisting of Kosovo's road network, Intelligence Preparation Battlespace (IPB) data and enemy doctrinal templates was constructed. This information was combined to generate an HMM capturing the specific situation.

The Traffic Generator provided an interface that allowed the user to script the behavior of one or more TELs. These TEL behaviors were then used to generate simulated Ground Moving Target Indication (GMTI) tracks. The user scripted TEL tracks were then merged with vehicle tracks corresponding to 300 unidentified (confuser) vehicles (see Figure 10.4).

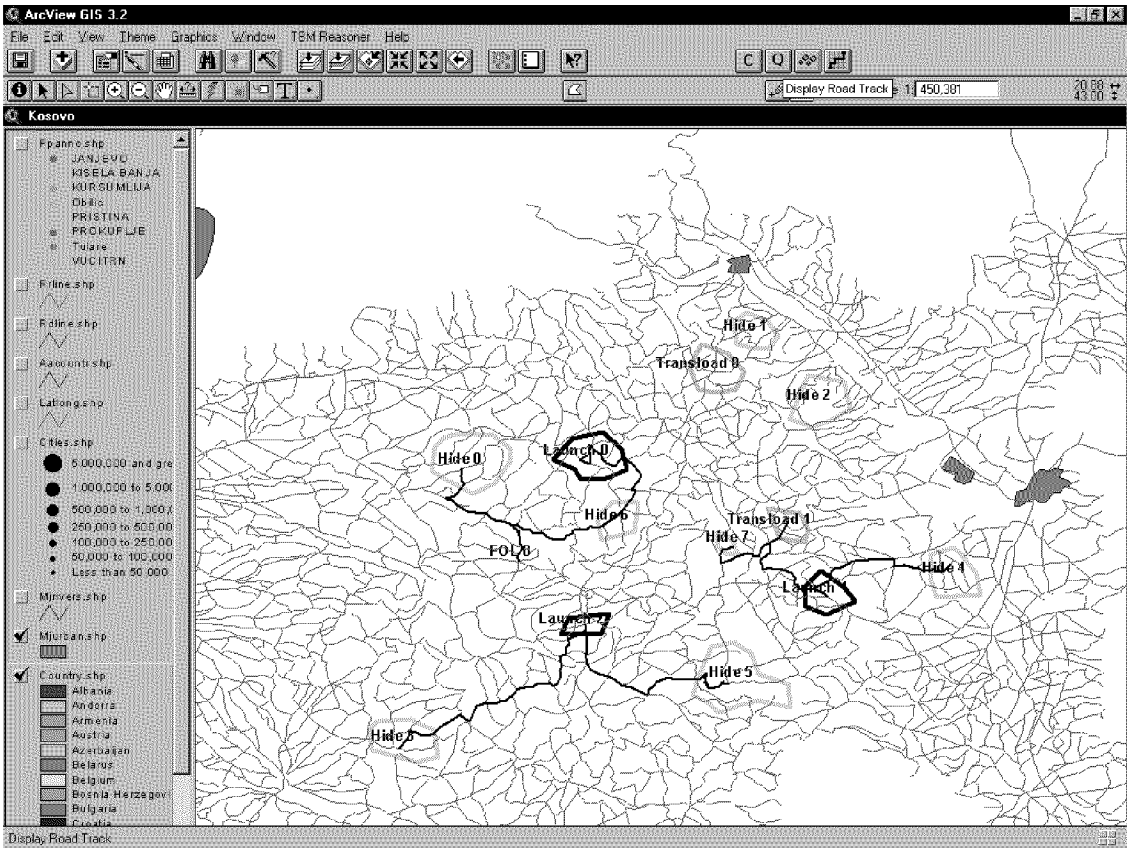

Figure 10.4. The Traffic Generator allows the user to generate TEL vehicle movements (tracks) in accordance with doctrinal TBM behavior templates.

The TBM Reasoner workstation accepted as input the collection of GMTI tracks (TELs and confuser vehicles) produced by the Traffic generator. The TBM Reasoner matched these tracks against the probabilistic model constructed for the Kosovo scenario and attempted to identify the tracks corresponding to the user scripted TEL tracks. The outcome of this effort was:

- A set of highlighted tracks corresponding to the suspected TEL vehicles
- Estimates of the suspected vehicle activities
- IPB data refinements based on the GMTI tracks of the identified vehicles

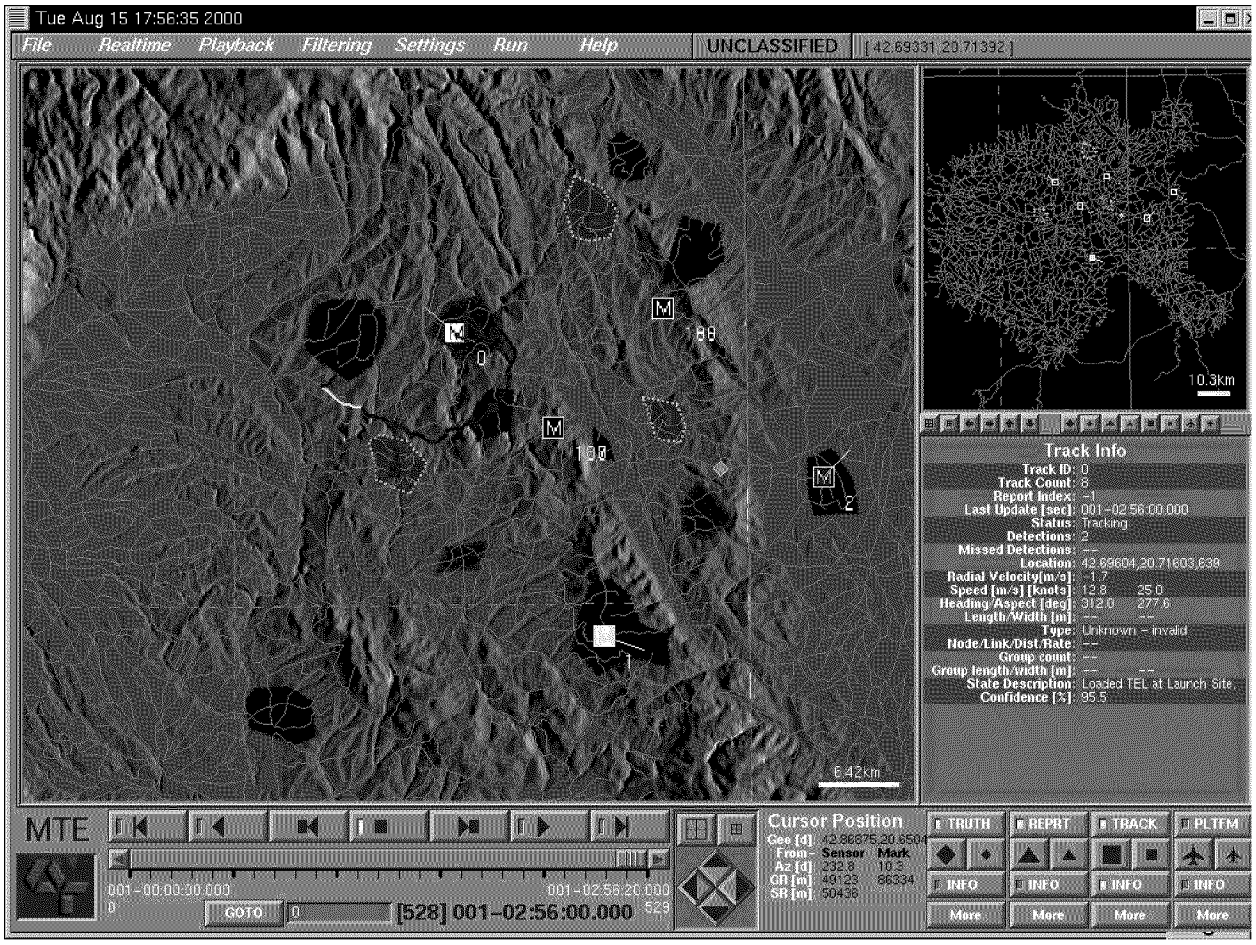

Figure 10.5. The TBM Reasoner detects and classifies TBM vehicles, reduces uncertainty in estimated location of suspected TBM sites and highlights the tracks corresponding to identified TEL vehicles.

A JEFX prototype was demonstrated during Joint Expeditionary Force eXperiment (JEFX) 2000 at Hurlburt Field, FL as a standalone Category III initiative (see Figure 10.5 depicting a prototypical screen). Personnel from ALPHATECH and the Air Force Research Laboratory (AFRL) briefed warfighters and answered questions on the initiative. The demonstration allowed operators to script TEL missions that the TBM Reasoner software then attempts to identify from simulated GMTI data.
As a standalone Category III initiative it proved to the reviewers to have "good potential for improving TCT planning and allocation of reconnaissance assets". In particular:

> "The TBM Reasoner improves the TCT cell's situational awareness of TBM vehicles and operating sites. It utilizes GMTI data to focus the analyst's attention on suspect vehicles and sites, improving the confidence in detecting and killing TCT's. The information produced by the application could be used to direct reconnaissance assets and alert strike packages of possible active threat locations."

11. JBI Study Task

11.1 Problem Overview

The JBI will provide smart middleware connecting information providers with information users in a Web-based publish-subscribe framework. To help subscribers identify potentially relevant information, domain ontologies for content tagging will need to be developed, encoded using an XML-based structured common representation (SCR). We envisage that the JBI will incorporate the following technologies:

1. *Knowledge authoring tools* to allow a community of interest to rapidly develop ontologies for content-tagging of documents and other information sources in a particular domain.

2. *Smart search engines* that exploit content tags on information sources to enable users to search for relevant information, and to filter and rank the results of searches.

Supporting distributed, collaborative development of ontologies for content tagging will be a central challenge for the JBI. The Web has proved successful in large part because information providers can deploy new content in a completely decentralized fashion, without the need to coordinate with one another. If the JBI's approach to content tagging is to succeed, it must likewise facilitate distributed development of ontologies by minimizing the need for coordination between ontology developers.

11.2 Objectives

To address this challenge, the goal is to develop preliminary concepts for ontology authoring to (a) allow different ontology developers whose domains of interest overlap to coordinate their ontology development efforts, and (b) enable smart search for information sources tagged according to an ontology that differs from the one used to construct the search query.

11.3 Approach

ALPHATECH specified preliminary designs for:

- **Fuselet-based *ontology connector services*.** An ontology connector service supports lightweight translation of representations in ontology x into equivalent or near-equivalent representations in a second ontology y. Such a service enables a smart search engine to employ queries encoded using ontology x to identify information sources that are content-tagged using elements of y. ALPHATECH is designing technology for (a) rapid authoring of ontology connector services by content providers, and (b) deploying these ontology connector services as fuselets in the JBI.

- **Distributed ontology refinement.** Once an ontology connector service $\tau(x,y)$ has been built linking ontology x to ontology y, it is natural to expect that the developers of x and y will continue to separately extend and/or modify their ontologies. ALPHATECH's distributed ontology refinement technology will facilitate maintenance of $\tau(x,y)$, both minimizing the extent to which the developers of x and y

need to coordinate with one another to maintain $\tau(x,y)$, and automatically identifying changes to x and y for which coordination is essential.

11.4 Results

These designs built upon technology currently under development in two AFRL initiatives: *An Instructable Agent for Rapid Knowledge Base Development* (a SBIR linked to DARPA's Rapid Knowledge Formation program), and Adaptive Sensor Fusion (ASF). Under the SBIR, ALPHATECH is developing technology allowing users who are not knowledge engineers to rapidly construct ontologies. Under ASF, ALPHATECH has developed a *Transport Knowledge Toolkit* (TKT) that provides transport-level connector services. ALPHATECH's tools for authoring ontology connector services and for supporting distributed ontology refinement will exploit the SBIR's ontology development technology (adapted to the SCR) to enable development of ontology translation services by information providers who are not also knowledge engineers. The TKT's transport-level connectivity services are then exploited to enable deployment of ontology connector services as fuselets in the JBI.

This preliminary design work performed under HPKB has led to a follow-on project funded by AFRL's "Fuselets" PRDA, which is currently realizing aspects of the design sketched here.

12. JBI Integration Task

12.1 Problem

Recently, the Air Force Scientific Advisory Board (AFSAB) has recommended that the Air Force adopt as a goal the development of a Joint Battlespace Infosphere, a military information management system for providing integrated mission understanding, shared awareness, shared planning, shared execution, shared visualization and shared predicted views. However, in doing so, they also recognized that currently deployed military information systems are unable to meet these requirements, because they employed "closed" architectures that rely upon fixed data flows between predetermined "stovepipe" components. It is inherently difficult to modify and extend the processing configuration of these architectures, making it very difficult to meet evolving mission requirements or to adapt to a rapidly changing operational context. Typically, many months, or even years, are required to make the engineering and software modifications necessary to change the data flow of such systems. Integration of the stovepipe systems with new components that were not explicitly accounted for during the design process is difficult, and often impossible.

12.2 Objectives

The objective of this effort was to develop a combat information management system that provides individual users with the specific information required for their functional responsibilities during crisis or conflict. In a joint effort between AFRL, Navy and Boeing, an initial JBI demonstration system was proposed which would used technology currently used in the Internet to overcome this problem. The JBI demonstration system integrates data from a wide variety of sources, aggregates this information, and distributes the information to subscribed users. In addition, the JBI platform integrates many individual information fusion systems that are designed to support operational forces. ALPHATECH's role was to adapt the TBM Reasoner to participate and integrate with the JBI demonstration system.

12.3 Approach

The approach adopted for this effort was to re-use as much of the technology and software components that were developed for the TBM Reasoner. Specifically, the model compilation software component was re-used to compile key products of this IPB process, including the battlespace environment, battlespace effects, prior intelligence regarding key enemy sites and activities, into detailed probabilistic state transition models encoded as Hidden Markov Models (HMMs). The pattern recognition engine was then used to automatically match the HMMs against track data to identify suspicious patterns of vehicle activity.

12.4 Results

The main activities for the JBI-CRADA integration effort were:

1) Participated in the scenario development
2) Designed an XML format to present the TBM Reasoner results
3) Compiled models for the JBI-CRADA scenario
4) Processed truth track data provided by AFRL.

Participated in the scenario development

Personnel from ALPHATECH Inc. participated during the scenario development effort which was held at Rome, NY. The result of the meeting was the generation of a scenario timeline.

Designed an XML format to present the TBM Reasoner results

In order to integrate with the JBI demonstration system, ALPHATECH developed an interface document defining the output records of TBM Reasoner and a schema for processing the data. It was decided that the TBM Reasoner will adopt an XML format to output its results.

Compiled models for the JBI-CRADA scenario

For the purposes of the demo, ALPHATECH developed knowledge base representations of the battlespace from Intelligence Preparation of the Battlespace (IPB) data, single TBM vehicle doctrinal templates of behavior and road network data. These representations provided a mechanism to refine the IPB areas of interest, and identify physical patterns of single vehicle activity within sets of intelligence data provided by level 1 trackers.

Knowledge representations were developed for the battlespace environment (including the road network), IPB areas of interest, enemy doctrinal templates (both *branches and sequels templates* and *spatial deployment templates*), and other IPB products (including terrain limitation data, prior intelligence, and enemy combat power estimates). Branches and sequels templates were used to specify doctrinal patterns of activity for small TBM units and key vehicles in those units, including temporal constraints. Spatial deployment templates provided additional spatial constraints on the layout of TBM units and key sites. Terrain limitation information provided additional constraints on possible locations of key sites and possible routes for TBM vehicles. Prior intelligence information included the locations of known and suspected TBM sites. The product of this process is a set of detailed probabilistic state transition models encoded as Hidden Markov Models (HMMs).

Processed truth track data

For the purposes of this demo, ALPHATECH Inc. was provided with a set of tracks and was tasked to identify the subset of tracks belonging to TCTs. This task was achieved by applying the pattern recognition engine, to matches the previously complied HMMs against sequences of observations abstracted from the track data. The pattern recognition engine determined (a) the sequence of track data segments that best matches the HMM (in the sense of yielding the highest posterior probability that the HMM applies given the track data sequence) and (b) the sequence of corresponding HMM states with highest probability. The output of this task is a sequence of TBM Reasoner reports (in XML format) that provide functional information about the identified TBM vehicles. This capability was delivered to AFRL and Boeing for integration into the JBI demonstration suite.

13. Summary

The HPKB Battlespace Challenge Problem program has served numerous purposes:

- demonstration and validation of AI and related information technologies to addressing military needs in planning, assessment, operations and intelligence
- assessment of the maturity of various AI and related information technologies, and providing associated guidance on future direction for research
- awareness of operational user community to the potential of such technologies and the opportunities for technology transition
- validation of the "Challenge Problem" framework/organization for future R&D programs.

While the formal DARPA HPKB program has concluded, the initiative continues to move forward with the use of its products (component technologies, integrated systems, assessments and demonstrations). New DARPA programs, such as RKF (Rapid Formation of Knowledge) are building upon the components and lessons-learned of HPKB; other service programs have focused on integrating technology products into existing operational systems. We envision such follow-on initiatives to continue for many years adding to the success of this program.